EIGENVALUE MINIMIZATION TECHNIQUES FOR DESIGNING OPTIMUM FIR COMPACTION FILTERS

Bogdan DUMITRESCU, Corneliu POPEEA

Department of Automatic Control and Computers
"Politehnica" University of Bucharest
313, Spl. Independenţei, 77206 Bucharest, Romania
e-mail: bogdan,popeea@schur.pub.ro

***Abstract**: In this paper we propose a new technique for finding the optimum FIR compaction filter adapted to signal statistics. The main novelty of our approach is the transformation of the original problem into the maximum eigenvalue minimization of a parameterized Toeplitz matrix, with a low number of variables. This is a typical application of semidefinite programming and may be solved with reliable interior-point algorithms. The optimal filter is then found either solving a quadratic system with a Newton-Raphson algorithm, or via a matrix Riccati equation. The numerical experiments show that the optimal compaction filter is obtained with good numerical accuracy and affordable execution time for filters of order up to 100. A characterization of optimal filters is also given, coherent with our matrix formulation of the optimization problem.*

1. INTRODUCTION

Filter banks adapted to signal statistics have received a large interest in the latest years due to their capability of compacting the energy of the input signal at the output of the first few channels (by selecting and ordering the principal components). As information from narrowband signals is easily extracted by filter banks, their applications in coding and compression are straightforward. The benefit is greater for signals with multiband structure, where classical designs prove to be less efficient. For its ease of implementation and stability properties, the class of FIR filters is of particular importance and our paper will be confined to it.

An optimum compaction filter maximizes the energy of its output, for the given class of input signals. For two-channel filters, with optimal subband bit allocation, an optimum compaction filter ensures also the optimality of the overall coding gain. In the general case, the optimum coding gain may not be obtained with FIR filters.

For a historical overview of proposed tech-

niques, we refer to [6]. We outline here the most recent approaches, giving also other details when the context will require it. As initially posed (see problem (H) in section 1), finding the optimal compaction filter is a nonconvex optimization problem which can be dealt with reliably only for very small orders. A turnout was produced by the approach via the product filter, which yielded several algorithms. Semi-infinite programming (SIP) was used by Moulin *et al* [10] as a way of approximating the solution. The same technique was adopted in the m-channel case [11], where once the filter for the first channel is obtained, the others are directly deduced in a paraunitary polyphase representation of the filter bank (for two-channel filter banks, the second filter results trivially from the first). Opposite to SIP methods, which are general, suboptimal and iterative, Kiraç and Vaidyanathan [6] designed an analytical method, which finds the optimal filter by a direct algorithm, but only for some classes of input signals, described by the authors as "lowpass", although more general than the name suggests. In the same paper, another fast algorithm is given, the window method, equivalent to a particular case of SIP. Finally, Tuqan and Vaidyanathan [17] transform the problem into a semidefinite program and find an optimal solution; as we will see in the sequel, their method has the drawback of a large number of variables.

The main novelty of our approach is a transformation of the problem, naturally leading to eigenvalue minimization. This is a particular but very significant case of semidefinite programming (SDP), i.e. an optimization problem where the objective function is linear and the constraints are expressed by a linear matrix inequality (LMI). In our case the number of variables is very low and the parameterized matrix has a Toeplitz structure. Typically, SDP problems are solved with interior-point methods, which have worst case polynomial complexity and a very good practical behavior; the primal-dual interior-point algorithms offer also a "certificate" of optimality, i.e. indicate the tolerance within the (global) optimum is attained. We may affirm that SDP is the major advance in optimization in the latest decade, combining powerful theoretical results with highly efficient computational techniques.

The recent literature on SDP is immense. Among the numerous papers, comprehensive reviews are given by Vandenberghe and Boyd [18] and, specifically for eigenvalue optimization, by Lewis and Overton [8]; interesting aspects of this latter problem are treated also by Alizadeh [2]. Apparently, SDP was seldom used in signal processing. For the first time, the paper of Tuqan and Vaidyanathan [17] uses SDP for solving the optimum compaction filter problem. Noll [12] uses eigenvalue optimization for spectrum estimation and image restoration. Related work is based on nonlinear convex optimization, see e.g. the paper of Wu, Boyd and Vandenberghe [20] on FIR filter design subject to bounds on the frequency response magnitude.

The content of our paper is as follows. After a presentation of the problem, we outline in section 2 the ideas of Tuqan and Vaidyanathan [17]. Section 3 is the kernel of our paper; it presents the transformation of the initial problem to a Toeplitz eigenvalue minimization problem. Three algorithms for computing the optimal compaction filter are deduced in section 4 and 5 and commented in detail in section 6. Section 7 is dedicated to a discussion of the structure of the optimal filters (i.e. the roots on the unit circle). Some numerical considerations and experiments conclude the paper.

2. BASIC DEFINITIONS AND NOTATIONS

Let $x(t)$ be a discrete time wide sense stationary stochastic signal with autocorrelation sequence defined by $r(k) = E[x(t)x(t-k)]$; by normalization we may take $r(0) = 1$ without loss of generality. The associated autocorrelation matrix R truncated at size $N+1$ is Toeplitz symmetric with elements on diagonal k equal to $r(k)$ (the main diagonal has number 0). Accordingly, in the sequel we will write $R = \mathrm{Toep}(r(0), r(1), \ldots, r(N))$. Let m be the number of channels of the FIR filter bank and suppose that $N+1$ is taken as a multiple of m, i.e. $N+1 = m(n+1)$.

Let $H(z)$ be a FIR filter of order N, with impulse response $h(k)$, $k = 0 : N$, that is $H(z) = \sum_{k=0}^{N} h(k)z^{-k}$; we also think at the sequence h as a unit norm vector in the familiar Euclidian space $\mathbb{R}^{N+1}$. The associated product filter is $G(z) = H(z)H(z^{-1}) = \sum_{k=-N}^{N} g(k)z^{-k}$. Obviously, we have $G(\mathrm{e}^{j\omega}) = |H(\mathrm{e}^{j\omega})|^2 \geq 0$ and g is a symmetric sequence, i.e. $g(k) = g(-k)$. The filter $H(z)$ is called a *compaction* filter if $G(\mathrm{e}^{j\omega})$ is Nyquist(m), i.e. $g(km) = \delta(k)$.

Conversely, a given product filter $G(z)$ corresponds to a (valid or feasible) compaction filter if the Nyquist(m) property is satisfied and in addition $G(\mathrm{e}^{j\omega}) \geq 0$.

The relation between the impulse responses g and h is immediate from the definition of the product filter. Denote $\Theta_k = \mathrm{Toep}(e_{k+1})$, $k = 0 : N$, where e_k is the unit vector of index k. Clearly, $g(0) = h^T h = \|h\|^2 = 1$, and for $k \neq 0$ we have

$$g(k) = \sum_{\ell=k}^{N} h(\ell)h(\ell - k) = \frac{1}{2} h^T \Theta_k h.$$

Since the Nyquist(m) sequence g has some elements equal to zero and $g(0) = 1$ is fixed, we will abusively denote g the vector containing only the non-trivial elements of the sequence, i.e.

$$g = [g(k)]_{k \in \mathcal{N}},$$

where

$$\mathcal{N} = \{k \mid 1 \leq k \leq N, k \bmod m \neq 0\}.$$

When $m = 2$, then $g = [g(1)\ g(3)\ \ldots\ g(N)]^T$. By putting $n' = n + 1$, the size of the vector g is $n'' = |\mathcal{N}| = N + 1 - n' = (m-1)(n+1)$, where $n' + n'' = N + 1$.

For a filter H and a given class of input signals described by the autocorrelation matrix R, the variance of the output signal $y(z) = H(z)x(z)$ is

$$\begin{aligned} \rho_y &= h^T R h = \sum_{k=-N}^{N} g(k)r(k) \\ &= r(0) + 2\sum_{k \in \mathcal{N}} g(k)r(k). \end{aligned} \tag{1}$$

A filter $H(z)$ maximizing the variance in (1) is called *optimum compaction* filter. There are two basic formulations of the problem of finding such a filter.

We may consider the unknown to be the impulse sequence h, case in which the problem is

$$\text{(H)} \quad \begin{array}{lll} \rho^o = & \max & h^T R h \\ & \text{subject to} & h^T h = 1 \\ & & h^T \Theta_{mk} h = 0, k = 1 : n, \end{array}$$

where ρ^o is the optimum compaction gain. The constraints $h^T \Theta_{mk} h = 0$ are also called orthogonality conditions. This is a quadratic optimization problem with n' nonconvex quadratic constraints and may be solved directly by classical constrained optimization methods, as in [19]. However, there is no way of guaranteeing global optimality of such a solution and difficulties in satisfying the complicated quadratic constraints are increasing as N grows larger.

An equivalent formulation may be stated in terms of the product filter, as in problem (G-SIP) shown at the top of next page (we remind the assumption $r(0) = 1$).

We have here a semi-infinite linear programming (SIP) problem, where the objective function and the constraints are linear, the number of variables is finite (and equal to n''), but the number of positivity constraints is infinite. The problem is convex and may be solved considering only a finite number of constraints, i.e. by discretizing the positivity constraints on a finite set $\omega_i \in [0, 2\pi)$, and using linear programming methods, as suggested first in [10]. Special care need to be taken to always ensure the positivity of g, using for example zero clustering [10] or windowing [6]. Of course, the obtained solution is suboptimal by its nature; near optimality is reached for finer discretization, increasing the number of constraints.

Although satisfactory compaction filters may be usually obtained by methods derived from the above two formulations, there is still no method to ensure optimality in all situations, regardless of input data and size of the problem. In our search, we have found necessary to re-formulate the problem.

$$\text{(G-SIP)} \qquad \begin{array}{rll} \rho^o = & \max & 1+2\sum_{k\in\mathcal{N}} r(k)g(k) \\ & \text{s.t.} & G(\mathrm{e}^{j\omega}) = 1+2\sum_{k\in\mathcal{N}} g(k)\cos k\omega \geq 0 \\ & & \forall \omega \in [0, 2\pi) \end{array}$$

3. A FIRST SDP APPROACH

Tuqan and Vaidyanathan [17] transformed (G-SIP) into a semidefinite programming (SDP) problem using the discrete time Kalman-Yakubovich-Popov (KYP) positivity lemma. Since this is the starting point of our approach, we present in some detail their formulation, with minor changes.

Consider the causal part of $G(z)$

$$G_+(z) = \frac{1}{2} + \sum_{k\in\mathcal{N}} g(k) z^{-k}. \tag{2}$$

The equivalence

$$G(\mathrm{e}^{j\omega}) \geq 0 \;\Leftrightarrow\; \mathrm{Re}G_+(\mathrm{e}^{j\omega}) \geq 0,$$

is obvious, so the positivity constraint of G is transferred to the condition of real-positivity of G_+. The (N-dimensional) state-space realization of $G_+(z)$ in controllable form is

$$G_+(z) = \left[\begin{array}{cc} A & b \\ c^T & \delta \end{array}\right],$$

where

$$\begin{array}{ll} A = \left[\begin{array}{cc} 0 & 0 \\ I_{N-1} & 0 \end{array}\right], & b = e_1, \\ c^T = g^T C, & \delta = 1/2 \end{array} \tag{3}$$

and $C \in \mathbb{R}^{n''\times N}$ is a matrix that expands g to length N, inserting zeros in the positions km, $k = 1 : n$ (clearly, C is obtained by adding zero columns to the unit matrix).

The real-positivity lemma [5] states that the system (A, B, C, D) is passive (or equivalently its transfer function is real-positive) if and only if a symmetric matrix $P \geq 0$ exists such that

$$\left[\begin{array}{cc} P - A^T P A & * \\ C^T - B^T P A & (D + D^T) - B^T P B \end{array}\right] \geq 0,$$

where $*$ denotes a block obtained by symmetry. In our case the positivity condition is equivalent to

$$Z = \left[\begin{array}{cc} 1 - b^T P b & * \\ C^T g - A^T P b & P - A^T P A \end{array}\right] \geq 0. \tag{4}$$

We remark that if (4) holds, then the inequality $P \geq 0$ is automatically fulfilled as a consequence of Lyapunov's lemma, since in our case A is obviously stable. Accordingly, we may assess now that (G-SIP) is equivalent to the following problem

$$\text{(G-SDP)} \qquad \begin{array}{rll} \rho^o = & \max & 1+2\sum_{k\in\mathcal{N}} r(k)g(k), \\ & \text{s.t.} & Z \geq 0 \end{array}$$

where Z is given by (4). Since the objective function is linear and the constraint is a linear matrix inequality, this is a semidefinite (linear) programming problem. While the idea of transforming (G-SIP) to (G-SDP) is indeed valuable, one drawback is the very large number of variables; there are n'' scalar variables in g and $N(N+1)/2$ in P; this fact leads to near intractability when N becomes large; for $N = 50$, there are already more than 1000 variables.

4. A NEW SDP: EIGENVALUE MINIMIZATION

Our idea is to solve not (G-SDP) as it stands, but its dual, which is another SDP problem and which turns out to have a much simpler form after some nontrivial appropriate transformations to be described in this section, exploiting the hidden structure of (G-SDP).

We need first to express in detail the (matrix) coefficients appearing in the LMI (4). Let $E_{k\ell}$ be the symmetric matrix with the elements in positions (k, ℓ) and (ℓ, k) equal to 1 and all other elements equal to 0. Let $\Delta_{k\ell} = E_{k+1,\ell+1} - E_{k\ell}$. Then, taking into account (3), Z from (4) may be written as in relation (5), shown at the top of next page.

$$Z = \begin{bmatrix} 1 & * \\ C^T g & 0 \end{bmatrix} - \begin{bmatrix} b^T \\ A^T \end{bmatrix} P[b\ A] + \begin{bmatrix} 0 & 0 \\ 0 & P \end{bmatrix} = \begin{bmatrix} 1 & * \\ C^T g & 0 \end{bmatrix} - \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & P \end{bmatrix} \quad (5)$$

$$= E_{11} + \sum_{k\in\mathcal{N}} g(k)E_{k+1,1} + \sum_{k=1}^{N}\sum_{\ell=1}^{k} P(k,l)\Delta_{k\ell}$$

$$\text{(D-SDP)} \qquad \begin{array}{rll} \lambda^o = & \min & 1 + \mathrm{tr}E_{11}X \\ & \text{s.t.} & \mathrm{tr}E_{k+1,1}X = -2r(k),\ k \in \mathcal{N} \\ & & \mathrm{tr}\Delta_{k\ell}X = 0,\ 1 \le \ell \le k \le N \\ & & X \ge 0 \end{array}$$

The dual. Using (5), the dual of (G-SDP) has the standard form (D-SDP) [18], shown above, where the matrix variable X is symmetric $(N+1)\times(N+1)$.

Theorem 1 *The problems (G-SDP) and (D-SDP) have the same optimal value, i.e.*

$$\rho^o = \lambda^o. \quad (6)$$

Proof. According to Theorem 3.1 in [18], the relation (6) holds if feasible g and P exist such that $Z > 0$. This is obvious for $g = 0$ and e.g. $P = \mathrm{diag}(\alpha, \alpha^2, \ldots, \alpha^N)$, with $0 < \alpha < 1$. ■

In order to continue our deduction, we need some simple basic facts from linear algebra. We work in the natural SDP frame, which is the linear space of symmetric matrices $\mathcal{S}$; since we will refer to (5), the size of the matrices is $(N+1)\times(N+1)$. (When the size is not clear from the context, we will add an index to $\mathcal{S}$, e.g. $\mathcal{S}_p$ is the space of $p \times p$ matrices.) The dimension of this linear space is $\dim\mathcal{S} = (N+1)(N+2)/2$ and the matrices $E_{k\ell}$, with $1 \le \ell \le k \le N+1$ form the standard basis of $\mathcal{S}$. The scalar product is here $\langle X, Y\rangle = \mathrm{tr}XY$, $X, Y \in \mathcal{S}$, so $\mathcal{S}$ is an Euclidian space. Two matrices $X, Y \in \mathcal{S}$ are orthogonal (on each other) if their scalar product is zero, i.e. $\mathrm{tr}XY = 0$. Finally, $\mathcal{S}$ can be (partially) ordered by using the (convex) cone of symmetric positive semidefinite matrices.

The set of symmetric Toeplitz matrices forms a significant linear subspace $\mathcal{T} \subset \mathcal{S}$ and $\dim\mathcal{T} = N+1$. The standard basis of $\mathcal{T}$ consists of the matrices $\Theta_k = \mathrm{Toep}(e_{k+1})$, $k = 0 : N$.

Denote now $\mathrm{tr}_k Y$ the sum of the elements of $Y \in \mathcal{S}$ along the k-th diagonal, where 0 is the main diagonal; this is an immediate generalization of the usual trace of a matrix.

Fact 1: Consider the following linear subspace of $\mathcal{S}$

$$\tilde{\mathcal{T}} = \{Y \mid \mathrm{tr}_k Y = 0,\ \forall k \in 0 : N\}. \quad (7)$$

Then $\dim\tilde{\mathcal{T}} = N(N+1)/2$ and the matrices $\Delta_{k\ell}$, $1 \le \ell \le k \le N$ form a basis of $\tilde{\mathcal{T}}$.

Proof. Immediate, after remarking that any $Y \in \tilde{\mathcal{T}}$ has the form

$$Y = \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & P \end{bmatrix},$$

where $P \in \mathcal{S}_N$ is a matrix whose elements are determined starting from the first column downwards the diagonals.

Fact 2: The orthogonal complement in $\mathcal{S}$ of the Toeplitz symmetric matrices subspace $\mathcal{T}$ is

$$\mathcal{T}^\perp = \tilde{\mathcal{T}}. \quad (8)$$

Proof. Imposing $\mathrm{tr}XY = 0$ for any $X \in \mathcal{T}$ (expressed as $X = \sum_k \alpha_k\Theta_k$) results in the condition $\sum_k \alpha_k \mathrm{tr}_k Y = 0$, for any set of coefficients α_k, i.e. $Y \in \tilde{\mathcal{T}}$. Since $\dim\mathcal{T} = N+1$, a dimensionality argument completes the proof.

The eigenvalue minimization problem. We are now in situation to give our main result of this section.

Theorem 2 *The problem (D-SDP) is equivalent to*

$$\text{(D)} \quad \begin{array}{rl} \lambda^o = & \min\ \lambda \\ & \text{s.t.}\ \ X = \lambda I - R + \sum_{k=1}^{n} \mu_k \Theta_{mk} \geq 0 \end{array}$$

Proof. According to the second constraint of (D-SDP), the variable X is orthogonal on all the matrices $\Delta_{k\ell}$; hence, it follows from Fact 2 that X is a Toeplitz matrix, that is $X(k,\ell) = X(|k-\ell|)$. Thus, $\text{tr}E_{k+1,1}X = 2X(k)$ and so the remaining constraints reduce to $X(k) = -r(k)$, for $k \in \mathcal{N}$. Finally, the objective function is simply $\lambda = 1 + X(1,1) = 1 + X(0)$, therefore $X(0) = \lambda - 1$.

The above conditions say that, by exploiting the equality constraints of (D-SDP), the general symmetric matrix variable X from (D-SDP) is transformed into a symmetric Toeplitz matrix which has the diagonals with index multiple of m free, while the other diagonals k are fixed to the values $-r(k)$, that is

$$X = \lambda I - R + \sum_{k=1}^{n} \mu_k \Theta_{mk}, \tag{9}$$

where the scalar λ and the vector $\mu \in \mathbb{R}^n$ are the new variables to be used in the optimization process described by (D). ■

Remark 1: As easily seen, the problem (D) consists of the minimization of the maximum eigenvalue λ of the symmetric Toeplitz matrix

$$R(\mu) = R - \sum_{k=1}^{n} \mu_k \Theta_{mk}, \tag{10}$$

which linearly depends on the parameters μ. (If the orthogonality constraints are missing, i.e. $n = 0$, then (D) reduces to the classical problem of computing the maximum eigenvalue of R.) Like its equivalent (G-SDP), (D) is a SDP problem, but it has only n' variables and a remarkable Toeplitz structure. Among all the equivalent formulations listed in this paper, this is the problem with the smallest number of variables for arbitrary m and so (D) is clearly the most numerically tractable.

The second dual. As a SDP problem, (D) has a dual of its own, which may be seen as another version of (G-SDP). Noticing that the matrix coefficients in the LMI constraint of (D) are the unit matrix for λ and Θ_{mk} for μ and appealing to the same mechanism used to pass from (G-SDP) to (D-SDP), we obtain the problem

$$\text{(P)} \quad \begin{array}{rll} \rho^o = & \max & \text{tr}RZ \\ & \text{s.t.} & \text{tr}Z = 1 \\ & & \text{tr}\Theta_{mk}Z = 0,\ k = 1:n \\ & & Z \geq 0 \end{array}$$

where the notation Z is used intentionally to emphasize the equivalence of this problem with (G-SDP). A simple look shows that Z from (5) satisfies the constraints of (P); it is also possible to prove that the objective functions are identical, using Fact 2 and (5).

We may now see that the problem (P) is not only equivalent to (D-SDP) but also similar to our original problem (H) in a very precise sense. Clearly, if h is a solution of (H), then $Z = hh^T$ is a solution of (P), because this particular form of Z transforms (P) into (H), as $\text{tr}Rhh^T = h^TRh$ etc. Therefore, we obtain (H) by adding to (P) the nonconvex condition $\text{rank}Z = 1$. The formulation (P) is called *convex relaxation* of (H) [18]; such relaxations are used to find suboptimal solutions to hard nonconvex problems; in our case, optimality is preserved. (The term convex relaxation itself is explained by the fact that the set of matrices Z with $\text{tr}Z = 1$ is the convex hull of rank-1 matrices hh^T with $\|h\| = 1$, see [13].)

Recapitulation. We finally have obtained six different formulations of the same problem, i.e. (H), (G-SIP), (G-SDP), (D-SDP), (D) and (P), all having the same optimal value. (G-SIP) is the initial formulation is terms of the product filter. (G-SDP) is obtained via the Kalman-Yakubovich-Popov lemma and (D-SDP) is its dual. Simplifying (D-SDP) results in (D) and its dual (P). Finally, (P) may be seen as a convex relaxation of (H). Closing the equivalence circle is possible by the basic equivalence between (H) and (G-SIP).

Although there are certainly still interesting details to be unveiled about the connections between these formulations, we conclude that the most appealing for numerical computation on a large spectrum of input data is the formulation (D).

5. OPTIMUM COMPACTION FILTER

From now on we will refer to the matrices Z from (G-SDP) and X from (D) as having the values corresponding to optimality. That is, X, Z are feasible and $XZ = 0$, by virtue of the complementarity property [18]. Also $\lambda^o = \rho^o$ in (D) is the maximal eigenvalue of $R(\mu^o)$ when the optimum is attained. Of course, λ^o is the optimum compaction gain.

We know how to compute the optimum, but not yet the optimum compaction filter $H(z)$. Solving (D) furnishes directly the solution to (H) only in one case which will be explained immediately; in the other cases, the solution of (D) must be further processed and this section provides a first algorithm.

Let ν be the multiplicity of λ^o as eigenvalue of $R(\mu^o)$. Generally, $\nu > 1$, as minimization tends to coalesce several eigenvalues in λ^o, as stressed in [8]. We see from (9) that X and $R(\mu^o)$ share the same eigenvectors and also that 0 is an eigenvalue of X of multiplicity ν.

Theorem 3 *Let h be a solution of (H), i.e. the impulse sequence of an optimum compaction filter. Then h is an eigenvector of $R(\mu^o)$ corresponding to λ^o (and an eigenvector of X corresponding to 0).*

Proof. Since h is a solution of (H), we have $h^T R h = \rho^o = \lambda^o$ and $h^T \Theta_{mk} h = 0$. Multiplying in (9), we obtain

$$h^T X h = \lambda^o - h^T R(\mu^o) h = 0.$$

Since λ^o is the maximum eigenvalue of $R(\mu^o)$ (and 0 is the minimum eigenvalue of X), the theorem holds as a consequence of Rayleigh's principle. ■

Corollary 1: If $\nu = 1$, then h is uniquely defined and solving (D) yields directly the solution to (H). As eigenvector corresponding to a unique maximum eigenvalue of a Toeplitz matrix, this optimum compaction filter has all the zeros on the unit circle [9].

A first algorithm for the optimal compaction filter. If $\nu > 1$, let $V \in \mathbb{R}^{(N+1)\times\nu}$ be a matrix containing on columns a complete set of (normalized) eigenvectors of $R(\mu^o)$ associated with λ^o; of course, the eigenvectors are pairwise orthogonal. Since h is an eigenvector of $R(\mu^o)$, then $h = Vu$, with $u \in \mathbb{R}^\nu$ and $\|u\| = 1$.

Corollary 2: In order to find h in the general case $\nu > 1$, the vector u must be computed such that the constraints of (H) are satisfied, i.e.

$$u^T C_k u = 0,\ k = 1:n, \quad \text{with } \|u\| = 1, \quad (11)$$
$$C_k = V^T \Theta_{mk} V.$$

Regardless the value of ν, this system with n' equations (the normalization condition included) and ν unknowns is compatible, following the equivalence of (D) and (H) and the discussion above.

The system (11) has a nice structure, due to the properties of eigenvectors of symmetric Toeplitz matrices, see [4]. Specifically, let J be the exchange matrix with ones on the antidiagonal and zeros otherwise; a vector w is symmetric if $w = Jw$ and skew-symmetric if $w = -Jw$. Then $\lceil \nu/2 \rceil$ of the eigenvectors of $R(\mu^o)$ corresponding to λ^o may be chosen to be symmetric and $\lfloor \nu/2 \rfloor$ skew-symmetric, or the vice-versa. We may order the columns of the matrix V such that symmetric vectors come first; we denote $V = [V_s\ V_a]$ this ordering. Remark now that $w_s^T \Theta_k w_a = 0$ for any vectors w_s symmetric and w_a skew-symmetric. Accordingly, the matrices C_k from (11) have a two-block diagonal structure, i.e.

$$C_k = \begin{bmatrix} V_s^T \\ V_a^T \end{bmatrix} \Theta_{mk} [V_s\ V_a] = \begin{bmatrix} C_{k1} & 0 \\ 0 & C_{k2} \end{bmatrix} \quad (12)$$

The system (11) may be solved using the Newton-Raphson method. The Jacobian matrix may be easily computed using only matrix-vector multiplications. When $\nu \neq n'$, then the Newton direction is computed in least squares sense. If $\nu < n + 1$, although the number of operations is larger, this choice is safer than simply retaining from (11) only $\nu - 1$ equations and the normalization condition. Anyway, the cost of solving the system is small with respect to the cost of SDP.

Generically, the system (11) has $2^{\nu-1}$ solutions. The Newton-Raphson algorithm com-

putes only one filter h, the result being strongly dependent on the initialization. The other solutions correspond to changes in the phase of the filter, i.e. to the replacement of some zeros of $H(z)$ by their reciprocal with respect to the unit circle. If one desires specific phase properties, the zeros of $H(z)$ must be computed, replaced adequately and then used to re-obtain the filter coefficients. We even may refine the coefficients of the new obtained filter h, using again the Newton-Raphson algorithm, this time with the initial approximation $V^T h$. It should be noted that we work here with a polynomial of degree N and spectral factorization is not involved. On the contrary, all the alternative spectral factorization procedures, which will be discussed in the next section, operate on polynomials of degree $2N$, with several double roots on the unit circle, and not only have a larger complexity, but also are more sensitive to numerical errors.

6. RETRIEVING THE PRODUCT FILTER

All the procedures described in [6] and [10] compute the product filter $G(z)$ by various methods and then $H(z)$ using spectral factorization. We will show now how the product filter may be found in our approach, as an alternative to the basic method presented in the previous section. Apparently, the reduction from (G-SDP) to (D) seems to have lost the product filter g on the way (as well as the matrix P). However, there is a simple way of retrieving it.

Remark 2: The class of primal-dual interior-point methods (the most successful in practice at this moment) solve simultaneously a SDP problem and its dual. They furnish not only the solution of the problem, e.g. λ^o and μ^o for (D) and trivially the matrix X, but also a solution Z of the dual problem. To say more, the vault key of primal-dual interior-point methods is to iteratively better and better approximate the fundamental equality $XZ = 0$ (which implies interesting properties, e.g. the symmetric matrices X and Z commute and therefore share the same eigenvectors). In the sequel, we suppose that such a method was used to solve (D), therefore Z is known.

Theorem 4 *The coefficients of the product filter are*

$$c(k) = tr_k Z, \quad k \in \mathcal{N}. \tag{13}$$

Proof. Immediate, from (5). ■

Remark 3: Although the matrix Z is not uniquely determined (the problem (P) has not a unique solution: if Z is a solution, then the same is true for $Z + Y \geq 0$, where Y has the form described in Fact 1), any Z leads to a valid product filter. Usually (in nondegenerate cases), the product filter is unique; it is the matrix P from (5) which undertakes the non-uniqueness of Z; this is another consequence of the omnipresent Fact 2 and relation (5).

For future developments, let us partition the matrix Z according to (4), i.e.

$$Z = \begin{bmatrix} \rho & * \\ s & Q \end{bmatrix} \geq 0. \tag{14}$$

The following equalities result

$$\begin{aligned} 1 - b^T P b &= \rho, \\ c - A^T P b &= s, \\ P - A^T P A &= Q. \end{aligned} \tag{15}$$

Of course, solving for P the Lyapunov equation above (using only additions), and then finding c from the second equation (15) gives the same result as (13).

Other algorithms for the optimal compaction filter. After computing g from (13), the optimum compaction filter may be computed in several ways by spectral factorization methods. A short review may be found in [20], where several algorithms are listed. We will come back later on this subject.

For the moment, we are interested by the technique mentioned by Tuqan and Vaidyanathan [17], which uses a matrix Riccati equation; we will present it with the aim of deducing our own method to compute $H(z)$ directly from Z. For the sake of clarity, we split the vec-

tor h representing the coefficients of the compaction filter as follows

$$h = \begin{bmatrix} \delta_0 \\ c_0 \end{bmatrix} \tag{16}$$

where $\delta_0 = h(0)$ and the elements of c_0 are $h(k)$, $k = 1 : N$.

As widely known, if the causal part $G_+(z)$ of the product filter $G(z) = H(z)H(z^{-1})$ is given, then there exists a one to one correspondence between the spectral factors $H(z)$ of $G(z)$ and the symmetric real solutions P_0 of the discrete matrix Riccati equation

$$P_0 = A^T P_0 A + \frac{(c - A^T P_0 b)(c^T - b^T P_0 A)}{1 - b^T P_0 b}, \tag{17}$$

where the pair (A, b) comes from (3) and c was computed from (15); the filter $H(z)$, as described by (16), is given by

$$c_0 = \frac{c - A^T P_0 b}{\delta_0}, \quad \delta_0 = (1 - b^T P_0 b)^{1/2}. \tag{18}$$

The spectral factor of minimum phase corresponds to the minimal solution $P_0 \geq 0$ of the Riccati equation (17).

For purposes that will be soon evident, the relations (17), (18) may be written in the equivalent form

$$\begin{aligned} 1 - b^T P_0 b &= \delta_0^2 \\ c - A^T P_0 b &= \delta_0 c_0 \\ P_0 - A^T P_0 A &= c_0 c_0^T. \end{aligned} \tag{19}$$

(We could remark here, thinking at the relaxation from (P) to (H) and at the hypothetical relation $Z = hh^T$, that the right members of (15) and (19) are respectively Z and the outer product hh^T.)

We are now able to indicate how the compaction filter may be computed directly from Z. Denoting

$$\Pi = P - P_0 \tag{20}$$

and subtracting the corresponding equations in (15) and (19), a new system results

$$\begin{aligned} \rho + b^T \Pi b &= \delta_0^2 \\ s + A^T \Pi b &= \delta_0 c_0 \\ -\Pi + Q + A^T \Pi A &= c_0 c_0^T, \end{aligned} \tag{21}$$

connecting the blocks of Z from (14) with the elements of h from (16) by means of the matrix Π.

Theorem 5 *The optimum compaction filter (16) is given by*

$$c_0 = \frac{s + A^T \Pi b}{\delta_0}, \quad \delta_0 = (\rho + b^T \Pi b)^{1/2}, \tag{22}$$

where Π *is the solution of the Riccati equation*

$$\Pi = Q + A^T \Pi A - \frac{(s + A^T \Pi b)(s^T + b^T \Pi A)}{\rho + b^T \Pi b}. \tag{23}$$

Proof. The theorem is an immediate consequence of relation (21). ■

Remark 4: Of course, due to the structure of A and b, there are in (22) only few additions and a square root extraction. However, in solving the Riccati equation (22), the full spectral factorization machinery is naturally involved.

The phase properties of the filter $H(z)$ result by choosing the appropriate solution of the Riccati equation (23).

7. SUMMARY OF ALGORITHMS AND DETAILS

In the previous sections we have presented three methods for solving the optimum compaction filter problem. Firstly, we will give them a short algorithmic form appropriate to implementation. Then, we will refine it into a Matlab main program, as shown in Figs. 1 and 2. Finally, we will insist on some practical details.

All the three methods may be seen as two-step procedures. The first step is to solve the eigenvalue minimization problem (D). To this purpose, we used the SDPT3 package of Toh, Todd and Tütüncü [16], designed for general SDP problems and written in Matlab; there are several primal-dual interior-point methods implemented in this package; we used the implicit choice, based on the Nesterov-Todd direction [15].

Our first method, described in section 4, may be outlined as follows.

Algorithm 1:

1.1. Solve (D) $\Rightarrow \lambda^o, \mu^o$, and $R(\mu^o)$ from (10).

1.2.1. Compute the eigenvectors $V \in \mathbb{R}^{(N+1)\times\nu}$ of $R(\mu^o)$ corresponding to λ^o.

1.2.2. Compute C_k, $k = 1 : n$, and find a solution u to the system (11) using the Newton-Raphson method. The optimal compaction filter is $h = Vu$.

1.2.3. (optional) Find the minimum phase (or other desired) filter equivalent to h, by root computation and replacement by reciprocal.

The second method, presented in section 5, uses the result of (D) in order to compute the product filter.

Algorithm 2:

2.1. Solve (D) $\Rightarrow \lambda^o$ and Z from (5), written as (14).

2.2.1. Compute the product filter g from (13).

2.2.2. Compute h by spectral factorization of the product filter.

The third method obtains the optimum compaction filter by solving directly a matrix Riccati equation and was described in the final of section 5.

Algorithm 3:

3.1. Solve (D) $\Rightarrow \lambda^o$ and Z from (5), written as (14).

3.2.1. Solve for Π the Riccati equation (23).

3.2.2. Compute h from (22) (in the form (16)).

A more detailed view of these three methods is given by the Matlab prototype implementation from Figs. 1 and 2. All the methods are merged in a single function; `h1`, `h2` and `h3` are vectors containing the impulse responses of the optimum compaction filter designed with the three methods. In writing this program, our aim was to present the simplest but most explicit form, renouncing at or hiding some speeding-up tricks. We tried to keep an immediate correspondence between the names of the variables and the notations used throughout this paper.

The first step has effectively only one important line, namely the call of the function solving SDP problems in the SDPT3 package; for an explanation of the arguments see the user manual at [16]. There are some minor differences between the statement of the standard SDP problem in (D) and in [16] (which solves a maximization problem, while (D) is a minimization one); hence, the profusion of minus signs in our program. The cell `Theta` stores the unit matrix, which is the coefficient of λ in the LMI associated with (D), and the matrices Θ_{mk}, the coefficients of μ. The vector `f` contains the coefficients of the objective function.

```
function [h1,h2,h3] = ocfilter(r, m)
% r     - correlation vector
% m     - number of channels

N = length(r) - 1;
n = floor( (N+1) / m ) - 1;

% STEP 1:
% prepare parameters for SDP problem (D)
blk{1,1} = 'nondiag'; blk{1,2} = N+1;
R = - toeplitz(r);
f = - eye(n+1,1);
Theta{1} = - eye(N+1);
for k = 1:n
  rr = zeros(1,N+1); rr(m*k+1) = 1;
  Theta{k+1} = - toeplitz(rr);
end
X0 = 0.001 * eye(N+1); Z0 = X0;
mu0 = zeros(n+1,1);

% solve (D) (eigenvalue minimization)
[val, Z, mu, X] = sdp(blk, Theta, R, ...
                      f, X0, mu0, Z0);
```

Fig. 1. Matlab skeleton of our algorithms, step 1.

Algorithm 1: As the implementation is straightforward, let us mention some hidden details. The tolerance used to detect to multiplicity of the maximum eigenvalue λ^o must be larger than the accuracy of the SDP; we consider that 10^{-6} is a good tolerance, although it may seem too large. (SDP solution accuracy is often at square root of the machine precision; however, we appreciate that λ^o is much more accurately computed.)

To take advantage of the form (12), we need a routine to compute the symmetric and skew-symmetric eigenvectors of $R(\mu^o)$. A technique

was indicated in [3]. Interestingly, the Matlab function `eig` furnished frequently the desired vectors when $\nu \leq n'$. Ordering the eigenvectors is trivial.

For the step 1.2.2, solving the quadratic system (11), we used a simple version of the Newton-Raphson method with adaptive step length and random initialization which was fully satisfactory; in case of nonconvergence in 20 iterations, the algorithm was restarted. The stopping criterion was twofold: firstly, a progress smaller than 10^{-9} in reducing the error norm (in the equality constraints of in (11)); secondly, a value of this error less than 10^{-5}.

The time complexity of solving (11) is largely influenced by the multiplicity ν of λ^o, which may have any value from 1 to $N+1$ (this latter case appears when R is diagonal). The case $\nu = n'$ is occurring frequently and there is an intuitive explanation to this fact: there are n variables in the optimization problem (excepting λ itself) and thus n degrees of freedom that allow coalescence of the maximal eigenvalue. However, this situation is not generic. The case $\nu < n'$ has also significant occurrence. On the contrary, the case $\nu > n'$, corresponding to degeneracy (multiple solutions of (D)), may be easily created artificially but seldom appears for randomly generated data (two cases in thousands of runs).

Algorithm 2: The sensible point of this algorithm is the spectral factorization. The main numerical difficulty in spectral factorization is given by the presence of double roots on the unit circle, natural for the optimal solution of the compaction filter problem. Their identification and separation is the ordeal of any method. We will present in section 8 our numerical remarks.

A simple algorithm, used for instance by Moulin and Mihçak [11], belongs to Lang and Frenzel [7] and effectively computes the zeros of $G(z)$ and of its derivative, classifies them in order to find pairs on the unit circle and recomputes the spectral factor from the appropriate roots.

The other algorithm we used is based on solving the matrix Riccati equation (17). Then,

```
% STEP 2:
% Algorithm 1:
rr = r;                 % build Ropt
for k=1:n               % use (10)
  rr(m*k+1) = rr(m*k+1) - mu(k+1);
end
Ropt = toeplitz(rr);

% find eigenvectors of Ropt
% corresponding to lopt
[V,L] = eig(Ropt);
lopt = max(diag(L));
V = V(:, find(abs(diag(L)-lopt) < tol));
nu = size(V,2);

if nu == 1
  % optimal filter already found
  h = V';
else       % solve quadratic system
  % put first symmetric eigenvector
  V = orderV( V );
  for k=1:n
    C{k} = V' * Theta{k+1} * V;
  end
  h1 = newtraph(C);
end

% Algorithm 2:
% retrieve product filter
for k=1:N, if rem(k,m) ~= 0
  c(k) = trace(Z{1}, k);
end, end

% spectral factorization
h2 = factspec(c);

% Algorithm 3:
A = diag( ones(N-1,1), -1 );
b = eye(N,1);

% split Z as in (14)
rho = Z{1}(1,1);
s = Z{1}(2:N+1,1);
Q = Z{1}(2:N+1,2:N+1);

% solve Riccati equation (23)
Pi = dare(A, b, Q, rho, s, eye(N));

% compute h as in (22), (16)
d0 = sqrt(rho + Pi(1,1));
c0 = (s + [Pi(2:N,1); 0]) / d0;
h3 = [d0; c0]';
```

Fig. 2. Matlab skeleton, step 2.

relations (18) are used to obtain the optimum compaction filter.

Algorithm 3: The Riccati equation (23) is used here, so it is the moment to remark that the standard algorithm used in the Matlab function `dare` stops execution when finding zeros (i.e. eigenvalues of the associated symplectic pencil) on the unit circle. However, zeros are classified inside or outside the circle using the working precision, i.e. the circle is indeed a very thin belt. Actually, the selection criterion is that half of the zeros be inside the circle.

Double roots are computed usually inaccurately, so the danger of finding roots exactly on the unit circle is practically insignificant. While the value of the zeros is affected in computation, it seems that the double zeros theoretically on the circle migrate one inside and one outside, i.e. that symmetry is a robust property.

8. THE ZEROS ON THE UNIT CIRCLE

When the solution to problem (G-SIP) is not unique, it is known that some zeros of all optimal compaction filters have a fixed position on the unit circle. One can find a thorough presentation in [10], for the case $m = 2$, and in [11] for arbitrary m. We aim to give in this section a specific look at this problem, consistent to our approach, i.e. using the formulation (D).

We will use a simple property of Toeplitz matrices. In what follows, vectors and polynomials of same length are seen as identical.

Lemma: Let $\tilde{R}$ be a symmetric Toeplitz matrix (of size $N+1$) whose maximum eigenvalue λ^o has multiplicity ν. Then, the eigenvectors of $\tilde{R}$ corresponding to λ^o have $\nu_0 = N+1-\nu$ common zeros on the unit circle.

Proof. Let R_0 be the leading principal submatrix of size $\nu_0 + 1$ of $\tilde{R}$. Due to the interlacing property of eigenvalues of leading principal submatrices (of symmetric matrices), the maximum eigenvalue of R_0 is λ^o and is unique. Let h_0 be the eigenvector of R_0 corresponding to λ^o. Then, h_0 has all its zeros on the unit circle (see [9]).

We notice that the vector $\tilde{h} = [h_0^T\ 0]^T \in \mathbb{R}^{N+1}$ is an eigenvector of $\tilde{R}$ corresponding to λ^o (we have $\tilde{h}^T\tilde{R}\tilde{h} = h_0^T R_0 h_0 = \lambda^o$ and λ^o is the maximum eigenvalue).

Since $\tilde{R}$ is a Toeplitz matrix, any of its principal blocks of size $\nu_0 + 1$ is equal to R_0; hence, any vector of the form $[0\ h_0^T\ 0]^T \in \mathbb{R}^{N+1}$, where the number of leading zeros may take any value between 0 and $\nu - 1$, is an eigenvector of $\tilde{R}$. These ν vectors are obviously independent and they form a basis of the eigenvector subspace corresponding to λ^o. As polynomials, they are $H_0(z)$, $zH_0(z)$, $\ldots$, $z^{\nu-1}H_0(z)$; any linear combination of these vectors has the ν_0 zeros of h_0 on the unit circle. ■

Corollary: Taking now $\tilde{R} = R(\mu^o)$, the optimal Toeplitz matrix corresponding to the solution of problem (D), and using the above lemma, we conclude that the zeros (on the unit circle) of h_0 belong also to the optimal compaction filter, which is a linear combination of eigenvectors.

Remark 5: The notion of *minimal root set* (on the unit circle) may be defined similarly to [10], this time for arbitrary m, and consists precisely of the roots of $H_0(z)$. Also, some of the results there may be easily generalized. For example, as in [11], we can express the given autocorrelation sequence as

$$r(k) = -\sum_{\ell=1}^{\nu_0} \rho_\ell \cos(k\omega_\ell), \quad k \in \mathcal{N},$$

where ρ_ℓ are positive and $\mathrm{e}^{j\omega_\ell}$ are the roots of $H_0(z)$; this is the Carathéodory representation of the elements of the Toeplitz matrix $\lambda^o I - R(\mu^o)$, see [4].

However, like Moulin and Mihçak [11], we can only conjecture about the relation between the multiplicity ν and the unicity of the solution (in terms of the product filter). To be precise, our opinion is that the solution is unique if and only if $\nu \le n'$ (i.e. $\nu_0 \ge n''$).

Algorithmic implications: A tempting way of computing the optimum compaction filter is to obtain first h_0 (after solving (D), of course) and then a vector h_1 such that

$H(z) = H_0(z)H_1(z)$. The vector h_1 may be determined in at least two main ways.

Firstly, as the product filter $G(z)$ is available as indicated by Theorem 4, we can take advantage of the equality $G(z) = G_0(z)G_1(z)$, where $G_0(z)$ and $G_1(z)$ are the product filter associated with $H_0(z)$ and $H_1(z)$. As $G_0(z)$ is easily computed, $G_1(z)$ may be obtained by some polynomial division algorithm adapted to the symmetry of the problem. (If $\nu \leq n'$, one may even use only the zero coefficients of $G(z)$ in order to form a determined linear system with the coefficients of $G_1(z)$ as unknowns. For $m = 2$, an algorithm is proposed in [1].) Then, $H_1(z)$ is obtained by spectral factorization; although this is not guaranteed, usually $G_1(z)$ has no roots on the unit circle; this fact enhances significantly the numerical reliability of spectral factorization.

Secondly, we can design a Newton-Raphson method for a system similar to (11). We use the appropriate basis for the eigensubspace of $R(\mu^o)$ corresponding to λ^o, i.e. the $(N+1)\times\nu$ matrix

$$W = \begin{bmatrix} h_0(0) & 0 & 0 & \dots & 0 & 0 \\ h_0(1) & h_0(0) & 0 & \dots & 0 & 0 \\ h_0(2) & h_0(1) & h_0(0) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & h_0(\nu_0) & h_0(\nu_0-1) \\ 0 & 0 & 0 & \dots & 0 & h_0(\nu_0) \end{bmatrix}$$

which has as columns all the vectors $[0\ h_0^T\ 0]^T$. Since W is equally a convolution matrix, the relation $h = Wh_1$ holds. Using this relation to express the constraints of (H), we obtain the system

$$\begin{aligned} &h_1^T D_k h_1 = 0,\ k = 1:n,\ \text{with } \|h_1\| = 1, \quad (24)\\ &D_k = W^T \Theta_{mk} W. \end{aligned}$$

This is a compatible system and the matrices D_k are all Toeplitz of size $\nu \times \nu$. Due to this structure, if $\nu \leq n'$, it is clear now that only $\nu - 1$ of the matrices D_k (and the unit matrix from the normalizing condition $\|h_1\| = 1$) are linearly independent.

Reliability considerations: Despite their simple form, the two approaches taken in this section fail numerically for relatively small values of N. One of the causes seems to be the computation of h_0. The Lemma above leads to unreliable implementations because R_0 has often close largest eigenvalues. An alternative way is to use the matrix V (from Corollary 2, section 4), which has orthogonal columns; we use the (right) triangularization $V = LU$, where L is lower trapezoidal and U orthogonal; then, h_0 is the last column of L, after removing the first $\nu - 1$ zeros.

We anticipate the next section with some numerical considerations regarding the two above algorithms. As the polynomial division (or deflation) is also numerically unstable, the first method is clearly unreliable. For $m = 2$, it appears that $N + 1 = 30$ is the higher value for which uncorrupted results are obtained. For the second method, solving (24) proved to give results up to $N + 1 = 40$.

As they are useful only for small number of unknowns, we stop here our report on the methods based on the separate computation of h_0. We appreciate that they are unreliable due to (definitive) the computation of h_0 without any regard to the constraints of (H).

9. NUMERICAL RESULTS

Optimality: The methods presented in this paper are all optimal, in the sense that they find the optimal compaction filter for any given N, m and correlation sequence r. We have presented in another paper [14] some comparisons with other algorithms, such as the analytical [6] or the linear programming [10, 6] methods.

We implemented also the algorithm of Tuqan and Vaidyanathan [17], which solves a modified version of problem (G-SDP), avoiding the need of a spectral factorization, with the LMI constraint expressed like in (5) and which is also optimal. We will denote T this algorithm.

This section is dedicated mainly to other aspects, as the execution time or the accuracy of computation. For shortness, our algorithms are denoted A1, A2 and A3.

Execution time: SDP algorithms involve a large amount of operations per iteration. On the contrary, the number of iterations is re-

markably stable with respect to the size of the problem. Overall, SDP algorithms have a worst-case polynomial complexity.

Since the package SDPT3 is written in Matlab (with some few mex files), as well as our programs, an exact report of computation times is not significant. To give only a figure, for $N+1=100$, $m=2$, the execution time was little greater than one minute for Algorithm 1, and less than two minutes for Algorithms 2 and 3. The largest problem we solved had $N+1=200$ and took less than half an hour (sparse matrice storage is used for the LMI coefficients Θ_{mk} at such large sizes).

As N grows, the time for SDP grows somehow slower than the time needed by the solution of the Riccati equation. Anyway, the time required to solve the Riccati equation is a significant part of the total time, e.g. one third. On the contrary, solving the system (11) scales better and the corresponding time is almost always within 10% of SDP time.

To have a complexity comparison, we give now some times for algorithm T. We remind that this SDP problem has $n''+N(N+1)/2$ variables, i.e. significantly more than in (D). Using the same SDPT3 Matlab routine, the execution time was greater than 4 minutes for $N+1=30$ and near to 13 minutes for $N+1=40$ (and $m=2$). We can conclude that our algorithms are significantly faster.

Accuracy of computation: The spectral factorization method of [7] fails often, more frequently as N grows. The method is probably too ambitious, as it tries to classify exactly some quantities affected by numerical errors (double roots of a polynomial are computed usually at half of the machine precision). The successful use of this method in [11] is certainly due to the fact that, in the context of SIP methods which give suboptimal results, there are no (exact) roots on the unit circle.

The Riccati equation method has a more robust approach, as we suggested in the previous section and managed to achieve the factorization for all runs. In the sequel, the spectral factorization in Algorithm 2 is supposed to be obtained via the Riccati equation (17).

Table 1 shows how accurate the computed filter h satisfies the constraints of (H); we adopted a simple error measure

$$e=\left(\sum_{k=1}^{n}(h^T\Theta_{mk}h)^2\right)^{1/2}.$$

Table 1. Magnitude order of worst-case error in the orthogonality conditions for our algorithms.

	Method		
$N+1$	A1	A2	A3
10	10^{-9}	10^{-13}	10^{-13}
20	10^{-7}	10^{-10}	10^{-11}
40	10^{-7}	10^{-11}	10^{-12}
60	10^{-7}	10^{-11}	10^{-11}
80	10^{-6}	10^{-10}	10^{-10}
100	10^{-6}	10^{-9}	10^{-11}

The table presents the magnitude order of the error, in the worst case. We used AR(8) processes to generate the correlations and made 40 runs for $N+1\leq 40$ and 20 runs for the other values of N. Other few runs were performed for values of $N+1$ from 120 to 200, without noticeable change in the accuracy.

It is interesting that the Riccati equation (17) gives a somehow smaller accuracy than (23), remark that give an advantage to A3, which is the most accurate of our three methods. Anyway, the faster Algorithm 1 has a very convenient accuracy, situated at little less than half root of the machine precision, even for large values of N.

Finally, let us mention that, for filter lengths where the comparison was possible (i.e. where algorithm T had an affordable execution time), our algorithms and T gave practically filters with the same compaction gain and constraints were respected at similar accuracies.

Examples: We considered an input signal generated by an AR(4) process, with two pairs of complex poles of magnitude 0.95. Using our algorithms, we designed optimum FIR compaction filters with 30 and 100 taps; the magnitude of their frequency responses is drawn in Fig. 3 for $m=2$, and in Fig. 4 for $m=3$.

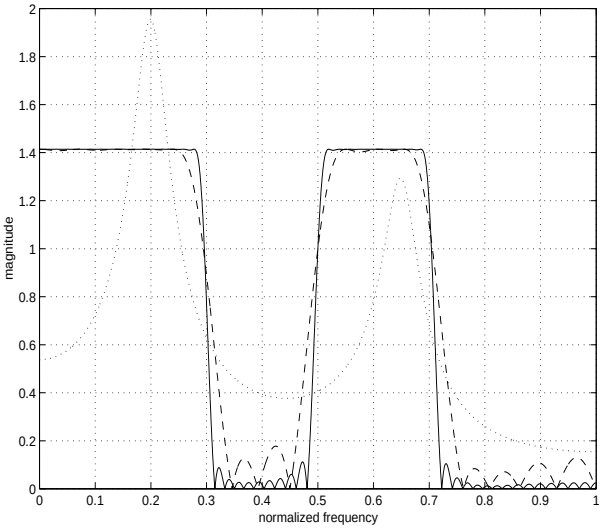

Fig. 3. Magnitude response of optimum compaction filter for an AR(4) process, for $N+1=30$ (dashed line) and $N+1=100$ (solid line), with $m=2$. The input signal spectral power is represented with a dotted line.

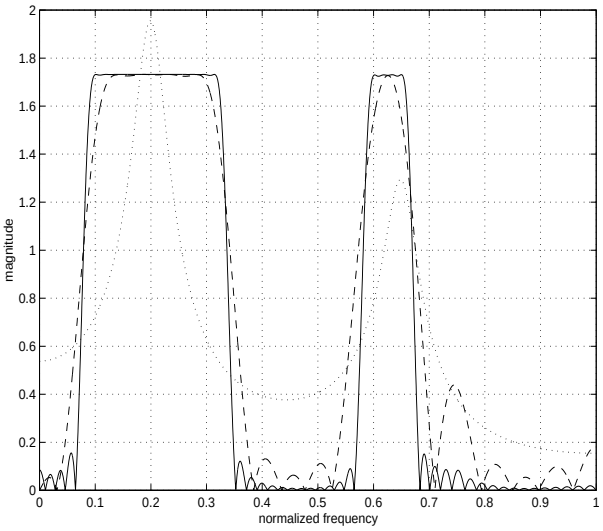

Fig. 4. Same as Fig.3, for $m=3$.

10. CONCLUSIONS AND FUTURE WORK

This paper was devoted to a new approach for computing the optimum compaction filter. The main idea is the reduction of the problem to the Toeplitz eigenvalue minimization (D), a semidefinite program with small number of variables. A second step is required, for which we have proposed three algorithms, one (A1) based on solving a quadratic system of equations and the others (A2 and A3) on different variants of spectral factorization.

Our interest in the subject remains open. There appear to be possibilities of generalization of some of the basic results in section 3 and connections with other problems may be made. Also, proving the conjecture in section 7 on the uniqueness of the solution for $m>2$ is also a significant aim.

11. REFERENCES

[1] Aas, K.C., Duell, K.A. and Mullis, C.T. "Synthesis of Extremal Wavelet-Generating Filters Using Gaussian Quadrature". *IEEE Trans. Signal Processing*, vol.43, no.5, pp.**1045–1057**, May 1995.

[2] Alizadeh, F. "Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization". *SIAM J. Optim.*, vol.5, no.1, pp.**13–51**, Feb. 1995.

[3] Cantoni, A. and Butler, P. "Properties of the Eigenvectors of Persymmetric Matrices with Applications to Communication Theory". *IEEE Trans. Communic.*, vol.24, no.8, pp.**804–809**, August 1976.

[4] Delsarte, P. and Genin, Y. "Spectral Properties of Finite Toeplitz Matrices". In *Mathematical Theory of Networks and Systems, Proc. MTNS-83*, pp.**194–213**, Beer Sheva, Israel, 1983.

[5] Ionescu, V., Oară, C. and Weiss, M. *"Generalized Riccati Theory and Robust Control: a Popov Function Approach"*. John Wiley and Sons, 1998.

[6] Kiraç, A. and Vaidyanathan, P.P. "Theory and Design of Optimum FIR Compaction Filters". *IEEE Trans. Signal Processing*, vol.46, no.4, pp.**903**–**919**, April 1998.

[7] Lang, M. and Frenzel, B.C. "Polynomial Root Finding". *IEEE Signal Processing Letters*, vol.1, pp.**141**–**143**, Oct. 1994.

[8] Lewis, A.S. and Overton, M.L. "Eigenvalue Optimization". *Acta Numerica*, vol.5, pp.**149**–**190**, 1996.

[9] Makhoul, J. "On the Eigenvectors of Symmetric Toeplitz Matrices". *IEEE Trans. Acoustics, Speech, Sign. Proc.*, vol.29, no.4, pp.**868**–**872**, August 1981.

[10] Moulin, P., Aniţescu, M., Kortanek, K. and Potra, F. "The Role of Linear Semi-Infinite Programming in Signal-Adapted QMF Bank Design". *IEEE Trans. Signal Processing*, vol.45, no.9, pp.**2160**–**2174**, Sept. 1997.

[11] Moulin, P. and Mihçak, M.K. "Theory and Design of Signal-Adapted FIR Paraunitary Filter Banks". *IEEE Trans. Signal Processing*, vol.46, no.4, pp.**920**–**929**, April 1998.

[12] Noll, D. "Reconstruction with Noisy Data: an Approach via Eigenvalue Optimization". *SIAM J. Optim.*, vol.8, no.1, pp.**82**–**104**, Feb. 1998.

[13] Overton, M.L. "Large Scale Optimization of Eigenvalues". *SIAM J. Optim.*, vol.2, no.1, pp.**88**–**120**, Feb. 1992.

[14] Popeea, C and Dumitrescu, B. "Optimal Compaction Gain By Eigenvalue Minimization". *Signal Processing*, vol.81, no.5, pp.**1113**–**1116**, May 2001.

[15] Todd, M.J., Toh, K.C. and Tütüncü, R.H. "On the Nesterov-Todd Direction on Semidefinite Programming". *SIAM J. Optim.*, vol.8, no.3, pp.**769**–**796**, August 1998.

[16] Toh, K.C., Todd, M.J. and Tütüncü, R.H. "SDPT3 – a Matlab Software Package for Semidefinite Programming". http://www.math.cmu.edu/∼reha/sdpt3.html.

[17] Tuqan, J. and Vaidyanathan, P.P. "Globally Optimal Two Channel FIR Orthonormal Filter Banks Adapted to the Input Signal Statistics". In *ICASSP*, vol.3, pp.**1353**–**1356**, Seattle, Washington, 1998.

[18] Vandenberghe, L. and Boyd, S. "Semidefinite Programming". *SIAM Review*, vol.38, no.1, pp.**49**–**95**, March 1996.

[19] Vandendorpe, L. "CQF Filter Banks Matched to Signal Statistics". *Signal Processing*, vol.29, pp.**237**–**249**, 1992.

[20] Wu, S.P., Boyd, S. and Vandenberghe, L. "FIR Filter Design via Spectral Factorization and Convex Optimization". In Biswa Datta, editor, *Applied and Computational Control, Signals and Circuits*, pp.**51**–**81**, Birkhauser, 1997.