IEEE TRANSACTIONS ON

AUTOMATIC CONTROL

A PUBLICATION OF THE IEEE CONTROL SYSTEMS SOCIETY

NOVEMBER 1997 VOLUME 42 NUMBER 11 IETAA9 (ISSN 0018–9286)

Now, the (i,j) entry of P (counting from zero) appears as $P_{ij} = s^t \mathcal{Z}^i (\mathcal{Z}^t)^j s$. As $\|\mathcal{Z}\| = 1$, the Cauchy–Schwartz inequality gives

$$|P_{ij}| \le \|(\mathcal{Z}^t)^i s\| \cdot \|(\mathcal{Z}^t)^j s\| \le \|s\|^2 = \sum_{k=1}^{\infty} s_k^2 = P_{00}$$

so that it suffices to examine the leading entry P_{00}. If we show that $\{s_k\}$ is absolutely summable ($\Sigma_k |s_k| < \infty$) whenever $H(z)$ is M-fold BIBO stable, square summability will follow as well.

Now, using the bound (27), the system $s = \Gamma_H c$ from the first column of (25) can be bounded componentwise as

$$\begin{aligned}
|s_1| &\le \beta(|h_1| + 2^{M-1}|h_2| + 3^{M-1}|h_3| + 4^{M-1}|h_4| + \cdots) \\
|s_2| &\le \beta(\quad\ |h_2| + 2^{M-1}|h_3| + 3^{M-1}|h_4| + \cdots) \\
|s_3| &\le \beta(\quad\qquad\ |h_3| + 2^{M-1}|h_4| + \cdots) \\
|s_4| &\le \beta(\quad\qquad\qquad\ |h_4| + \cdots) \\
&\ \ \vdots
\end{aligned}$$

The absolute sum of $\{s_k\}$ is then bounded as

$$\sum_{k=1}^{\infty} |s_k| \le \beta \sum_{k=1}^{\infty} \left(\sum_{i=1}^{k} i^{M-1} \right) |h_k|$$

where $\Sigma_{i=1}^{k} i^{M-1} \le k^M$. This gives

$$\sum_{k=1}^{\infty} |s_k| \le \beta \sum_{k=1}^{\infty} k^M |h_k|$$

in which the right-hand side remains finite whenever the functions $H(z), dH(z)/dz, \cdots, d^M H(z)/dz^M$ remain BIBO stable. This completes the boundary case, to prove Theorem 1.

IV. Concluding Remarks

Our main result shows that the Steiglitz–McBride algorithm does admit a stationary point in reduced-order cases, though the present result is restricted to white noise inputs and disturbances. Our interest in affirming existence stems from a previous result [7] showing an attractive error bound at any such stationary point. Whether the set of stationary points will always include an attractor point to either the on-line [3] or off-line [2] version is not revealed from our analysis.

One is tempted, of course, to extend this result to correlated inputs; let us pinpoint where the construct breaks down. For correlated inputs, one may still arrive at a matrix equation akin to (22), but the resulting P matrix is no longer close to Toeplitz in the sense of (26), i.e., its displacement rank increases. The coefficients $\{\alpha_k\}$ then need not yield the coefficients of a minimum phase polynomial (see e.g., [2]). The fixed-point theorem evoked here no longer applies, but this, of course, does not belie the existence of a stationary point for that case.

References

[1] K. E. Steiglitz and L. E. McBride, "A technique for the identification of linear systems," *IEEE Trans. Automat. Contr.*, vol. 10, pp. 461–464, Oct. 1965.

[2] P. Stoica and T. Söderström, "The Steiglitz–McBride identification algorithm revisted—Convergence analysis and accuracy aspects," *IEEE Trans. Automat. Contr.*, vol. 26, pp. 712–717, June 1981.

[3] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circuits Syst.*, vol. 33, pp. 939–947, 1986.

[4] H. Fan, "Application of B...aiste's convergence results in the study of adaptive IIR filtering algorithms," *IEEE Trans. Inform. Theory*, vol. 34, pp. 692–705, July 1988.

[5] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1983.

[6] P. A. Regalia, *Adaptive IIR Filtering in Signal Processing and Control*. New York: Marcel Dekker, 1995.

[7] P. A. Regalia and M. Mboup, "Undermodeled adaptive filtering: An *a priori* error bound for the Steiglitz-McBride method," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 105–116, Feb. 1996.

[8] H. Fan and M. Nayeri, "On reduced order identification: Revisiting 'On some system identification techniques for adaptive filtering,'" *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1144–1151, Sept. 1990.

[9] M. Mboup, "Filtrage adaptatif par structures prédictive et récursive: Application à l'annulation d'echo acoustique," Ph.D. dissertation, Université Paris-Orsay, Dec. 1992.

[10] J. L. Walsh, *Interpolation and Approximation by Rational Functions in the Complex Domain*. New York: Amer. Math. Soc., 1935.

[11] C. K. Chui and G. Chen, *Signal Processing and Systems Theory*. New York: Springer, 1992.

[12] T. L. Saaty and J. Bram, *Nonlinear Mathematics*. New York: McGraw-Hill, 1964.

[13] J.-P. Aubin, *L'Analyse non Linéaire et ses Motivations Economiques*. Paris: Masson, 1984.

[14] C. T. Mullis and R. A. Roberts, "The use of second-order information in the approximation of discrete-time linear systems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 24, pp. 226–238, June 1976.

Calculation of the Structured Singular Value with Gradient-Based Optimization Algorithms on a Lie Group of Structured Unitary Matrices

Jeroen Dehaene, Cheng Yi, and Bart De Moor

***Abstract*—The structured singular value problem, which is a basic problem in robustness analysis and design of multivariable controllers, can be formulated as an optimization problem over the manifold of unitary matrices with a given structure. We show how geometric optimization methods, such as the steepest ascent method and the conjugate gradient method for optimization on a Riemannian manifold, lead to algorithms giving a guaranteed nontrivial lower bound for the structured singular value.**

***Index Terms*—Gradient-based optimization, structured singular value (μ).**

Manuscript received September 26, 1994; revised February 29, 1996 and September 23, 1996. This work was supported by the Flemish Government: Concerted Research Action MIPS (Model-based Information Processing Systems), the FWO (Fund for Scientific Research-Flanders): G.0292.95 Matrix algorithms and differential geometry for adaptive signal processing, system identification and control, the FWO Onderzoeksgemeenschappen: ICCoS (Identification and Control of Complex Systems) and Advanced Numerical Methods for Mathematical Modeling, the IWT (Flemish Institute for Science and Technology in Industry): IWT/VCST project, Design of an electronically controlled Continuous Variable Transmission (1992–1997); the Belgian Government: Interuniversity Attraction Poles (IUAP-17 (1991–1995): Modeling and Control of Dynamical Systems; and IUAP-50 (1992–1996): Automation in Design and Production). J. Dehaene is a postdoctoral researcher of the Fund for Scientific Research Flanders (FWO).

J. Dehaene and B. De Moor are with ESAT-SISTA, K. U. Leuven, B-3001 Leuven (Heverlee), Belgium.

C. Yi is with Dynamic Control Systems Inc., Delta, B. C. V4G 1H5 Canada.

Publisher Item Identifier S 0018-9286(97)07639-3.

I. I...oduction

The structured singular value, generally referred to as μ, is an important tool for the robustness analysis of systems with a structured uncertainty [1]. In this section we formulate the definition of μ as an optimization problem and situate our optimization strategy and the existing power iteration algorithm (PIA) [2] in a more general scheme.

The structured singular value can be defined by (see [2] for equivalent definitions) the following.

Definition 1:

$$\mu_{\Delta}(M) = \max_{Q \in \mathcal{Q}} \rho(QM)$$

where $\rho(QM)$ denotes the spectral radius, that is, the maximal absolute value of an eigenvalue of QM.

$\mathcal{Q} = \{Q \in \Delta | Q^H Q = I_n\}$ is the set of *structured unitary matrices*, and Δ is a set of block diagonal matrices, defining the *structure*. $\Delta = \{\mathrm{diag}[\delta_1 I_{r_1}, \cdots, \delta_s I_{r_s}, \Delta_1, \cdots, \Delta_f] | \delta_k \in \mathbb{C}, \Delta_l \in \mathbb{C}^{m_l \times m_l}\}$.

The blocks $\delta_k I_{r_k} \in \mathbb{C}^{r_k \times r_k}$ are *repeated scalar blocks*.

Blocks of the form $\Delta_l \in \mathbb{C}^{m_l \times m_l}$ are *full blocks*.

We will refer to the submatrices of a matrix A with the same size and position as $\delta_k I_{r_k}$ and Δ_l, as $A_{r_k, r_k}, k = 1, \cdots, s$ and $A_{f_l, f_l}, l = 1, \cdots, f$, respectively.

Remark 1: All presented results also hold for nonsquare matrices M, by extending them with zero rows or columns to make all blocks square.

No algorithm is known that finds the global optimum of the optimization problem; instead, algorithms used today find lower-bounds for μ. This is also the case for the algorithms presented in this paper. Other algorithms exist that give upper-bounds, such that both together provide the user with an interval containing μ.

The PIA of Packard *et al.* [2] computes a lower-bound for μ and is computationally efficient if it converges. However, to date, there is no proved convergence result. Current implementations do not necessarily converge [2]–[4], although experiments show that the algorithm converges very often. If the algorithm does not converge, no lower-bounds result, except trivial ones. Nevertheless, in practice the PIA algorithm proves to be a very good engineering tool. The approach developed in this work is less efficient but has a less heuristic theoretical foundation. Both approaches fit into the following general discussion.

To solve an optimization problem (in the sense of finding a local maximum), one can either apply *general optimization algorithms* using function evaluations and possibly gradient information, to find a locally optimal solution, or one can concentrate directly on the *optimality conditions* for the given problem (taking different forms depending on the type of constraints and the smoothness of the problem).

If the *optimality conditions* cannot be solved analytically, an iterative scheme is needed to solve them. Often, it is not difficult to produce (many) iterative schemes which converge to solutions that satisfy the optimality conditions, if they converge at all. If computer simulations show that the scheme converges in a vast number of cases, such algorithms may be good engineering tools. This is where the PIA fits into the scheme. It converges in the majority of cases but without proof.

In this work we choose the approach of *general optimization algorithms*, more specifically the steepest ascent algorithm (SAA) and the conjugate gradient algorithm (CGA). Actually, such an approach may be regarded as just another iterative algorithm to satisfy the optimality conditions, but now one with interpretable intermediate results and provable convergence results (see below). The price for these results may be the loss of computational efficiency. We suggest that both approaches are combined. One could for instance first run the PIA and run one of the new algorithms if it fails. Or if one can afford the computational effort, it is always good to run different algorithms as these may give better lower bounds.

We will not rigorously prove convergence in terms of Q for the discrete-time algorithms. This would imply a detailed consideration (and possibly adaptation) of the line search algorithms to avoid jumping between different points Q with the same (or almost the same) value for the objective function ρ. However, since only the obtained value of ρ, and not the point Q where it is reached, is important, it is not worth putting too much energy in such considerations. We come back to the convergence issue at the end of Section III.

In the above discussion, we did not yet mention the constraint. To take into account the highly regular constraint $Q \in \mathcal{Q}$ of the optimization problem (Definition 1), we directly define an algorithm that "lives" in $\mathcal{Q}$ instead of in a Euclidean space. The steepest ascent and CGA's on this manifold will be derived in Sections III and IV. First, the next section introduces some facts about the manifold $\mathcal{Q}$.

This paper is further organized as follows. Section II discusses the Lie group of structured unitary matrices. In Sections III and IV the SAA and the CGA for the structured singular value problem are derived. Section V gives numerical experiments.

We also refer to [5], where some smooth approximations of the optimization problem are discussed, from which an initial value for the algorithms can be derived. In [5] and [3] we also discuss continuous-time algorithms.

II. The Manifold $\mathcal{Q}$ of Structured Unitary Matrices

The optimization methods discussed in [6] generalize optimization methods over Euclidean spaces to optimization methods over Riemannian manifolds by making use of intrinsic properties of the manifold. To this end, *straight lines are replaced by geodesics*. Tangent vectors at one point are adapted to other points by parallel translations along geodesics. In general, the calculation of geodesics and parallel translation on Riemannian manifolds is not easy. However, because the manifold $\mathcal{Q}$ of structured unitary matrices is a Lie group, simpler formulas can be obtained. The following theorem establishes the main properties of $\mathcal{Q}$ for later use. We refer to [5] for a proof.

Theorem 1:

1) The manifold $\mathcal{Q} = \{Q \in \Delta | Q^H Q = I_n\}$ is a Lie group.[1]
2) The tangent space at $Q \in \mathcal{Q}$ equals
$$T_Q\mathcal{Q} = \{T \in \mathbb{C}^{n \times n} | T = SQ, S \in \Delta, S^H + S = 0_n\}. \quad (1)$$
3) The inner product of tangent vectors $RQ \in T_Q\mathcal{Q}$ and $SQ \in T_Q\mathcal{Q}$, induced by the standard inner product in $\mathbb{C}^{n \times n}$ (identified with $\mathbb{R}^{2n^2}$), can be written as
$$\langle RQ, SQ \rangle = \langle R, S \rangle = \mathrm{Tr}(R^H S). \quad (2)$$
4) Given the inner product (2) and the corresponding Riemannian connection, the geodesic emanating from $Q = Q(0)$ in a direction SQ, parameterized by the real parameter t, is given by
$$Q(t) = \exp(St)Q. \quad (3)$$
5) The parallel translation of a tangent vector $RQ = R(0)Q(0) \in T_Q\mathcal{Q}$ along the geodesic (3) is described by
$$R(t)Q(t) = \exp(\tfrac{1}{2}St)R(0)\exp(-\tfrac{1}{2}St)Q(t). \quad (4)$$

[1] For background material about Lie groups we refer to [7] and [8].

III. The Steepest Ascent Algorithm

In this section we derive the SAA on the manifold $\mathcal{Q}$, for the optimization problem of Definition 1. First we introduce the algorithm, then we consider the computational cost. Finally, we discuss the convergence issue.

A. The Algorithm

The SAA is an iterative algorithm, calculating in each step $k = 1, 2, \cdots$ a new point $Q^{[k]}$ on the manifold $\mathcal{Q}$, given a point $Q^{[k-1]}$. To this end, a (one-dimensional) optimization problem, often referred to as *line search*, is solved, maximizing $\rho(QM)$ along a geodesic on $\mathcal{Q}$, emanating from Q_{k-1} in the direction of the gradient. To simplify some of the derivations, $\Phi(Q) = (\rho(QM))^2$ will be used as an objective function, instead of $\rho(QM)$.

The objective function $\Phi(Q)$ is not differentiable at all points $Q \in \mathcal{Q}$. Theoretically, three cases are to be distinguished.

Case a: The spectral radius $\rho(QM)$ corresponds to one eigenvalue of QM with algebraic multiplicity one.

Case b: The spectral radius $\rho(QM)$ corresponds to $k > 1$ eigenvalues of QM with algebraic multiplicity one.

Case c: The spectral radius $\rho(QM)$ corresponds to an eigenvalue with algebraic multiplicity greater than one.

Case a is generic. Case b is likely to occur at isolated points of some geodesics (when the absolute value of one eigenvalue crosses the absolute value of another eigenvalue). However, since $\rho(QM)$ is defined as $\max_i |\lambda_i|$, where λ_i are the eigenvalues of QM, and since these absolute values are smooth functions of Q in a neighborhood of points of Case b, the maximum of $\rho(QM)$ cannot occur at a point of Case b unless $\rho(QM)$ and $\Phi(QM)$ are differentiable at that point. Case c does in general not occur, and we will take a heuristic approach (see Remark 3). The following theorem gives a formula for the gradient in differentiable points and shows how the same formula gives an ascent direction (or zero in a local maximum) for points of Case b.

Theorem 2:

1) Let the spectral radius of $X = QM$ correspond to one eigenvalue λ of X with algebraic multiplicity one (Case a above), and let p be a corresponding right eigenvector and v a corresponding left eigenvector, satisfying $v^H p = 1$. The gradient of $\Phi(Q) = \rho^2(QM)$, at the point $Q \in \mathcal{Q}$, with respect to the standard induced inner product (2) is given by $\text{grad}_Q\ \Phi = \Phi(Q)GQ$, where $G = \pi(vp^H - pv^H)$, and π denotes the orthogonal projection on Δ and is given by $\pi(A) = \text{diag}[\alpha_1 I_{r_1}, \cdots, \alpha_s I_{r_s}, A_{f_1,f_1}, \cdots, A_{f_f,f_f}]$ for any $A \in \mathbb{C}^{n\times n}$, where $\alpha_k = (1/r_k)\ \text{Tr}(A_{r_k,r_k}), k = 1, \cdots, s$. That is, π zeros elements that have to be zero in Δ and averages elements that have to be equal in Δ.
2) If the spectral radius $\rho(QM)$ corresponds to $m > 1$ eigenvalues of QM with algebraic multiplicity one, then $\rho(QM)$ is the maximum of m differentiable functions, corresponding to the different eigenvalues with absolute value $\rho(QM)$. For any choice λ among these eigenvalues, and the corresponding right and left eigenvectors p and v satisfying $v^H p = 1$, $\Phi(Q)GQ$ is the gradient of one of these m functions and gives an ascent direction of $\rho(QM)$ or gives zero in a critical point.

Proof:

1) Clearly, $\Phi(Q)GQ$ belongs to the tangent space $T_Q\mathcal{Q}$, since $\Phi(Q)\pi(vp^H - pv^H)$ is skew Hermitian and belongs to Δ. The only thing left to prove is that the directional derivative of Φ at Q in a direction SQ is given by $\langle \text{grad}_Q\ \Phi, SQ\rangle$. Or equivalently, if $Q(t)$ describes some path in $\mathcal{Q}$ through $Q = Q(0)$, with velocity $\dot{Q}(0) = SQ \in T_Q\mathcal{Q}$ (thinking of t as time), and $\Phi(t) = \Phi(Q(t))$ that

$$\dot{\Phi}(0) = \langle \text{grad}_Q\ \Phi, SQ\rangle. \tag{5}$$

From $X = QM$ it is clear that $\dot{X} = \dot{Q}M = SQM = SX$. And since by assumption the spectral radius of X corresponds to only one eigenvalue λ, $\lambda(t)$ is differentiable at $t = 0$ and $\dot{\lambda}(0) = v^H\dot{X}p = v^H SXp = \lambda v^H Sp$ [9], [2]. Finally, $\dot{\Phi}(0) = (d/dt)(\lambda^*\lambda) = \dot{\lambda}^*\lambda + \lambda^*\dot{\lambda} = \lambda^*\lambda v^H Sp + \lambda^*\lambda p^H S^H v = \Phi(Q)(v^H Sp - p^H Sv) = \langle S, \Phi(Q)(vp^H)\rangle = \langle \pi(S), \Phi(Q)(vp^H - pv^H)\rangle = \langle S, \Phi(Q)\pi(vp^H - pv^H)\rangle = \langle SQ, \Phi(Q)\pi(vp^H - pv^H)Q\rangle$, which proves (5).

2) In a neighborhood of Q, $\Phi(Q)$ can now be regarded as the maximum of m functions that are equal at Q. The gradient of each of these functions separately is given by $\Phi(Q)GQ$, with one possible choice for λ, and the corresponding p and v. These gradients give steepest ascent directions (or zero in critical points) for the individual functions and therefore certainly give ascent directions for the maximum $\Phi(Q)$. One of the gradients corresponds to the steepest ascent direction. ∎

Application of the steepest ascent method to the present problem now yields the following algorithm.

Algorithm 1 (SAA):

1. initialize Q_0
2. $k = 0$
3. **repeat**
4. compute $\rho = \rho(Q_k M)$ and corresponding right and left eigenvectors p and v, such that $v^H p = 1$
5. $G = \pi(vp^H - pv^H)$
6. $t = \arg\max_{t>0}\ \rho(\exp(Gt)Q_k M)$
7. $Q_{k+1} = \exp(Gt)Q_k$
8. $k = k + 1$
9. **until** convergence.

Remark 2: The exponentiation of $\exp(Gt) = \exp(\pi(vp^H - pv^H)t)$ can be calculated from the block components separately. For each block an eigenvalue decomposition (of a skew symmetric matrix) must be computed (once per step). We refer to [5] for more details.

Remark 3: For points of Case c, calculation of an ascent direction is much more difficult than for other points. In addition, the effort is not worthwhile, as this case does not normally occur, and if it occurs it can generally be avoided. In most cases, when a point of Case c is met during the line search, one can simply take another point. In many cases points of Case c can be avoided by multiplying Q with $U \in \mathcal{Q}$, for which $Up = p$ and $U^H v = v$. In that case $\Phi(UQ) = \Phi(Q)$, but in general UQ will no longer be of Case c. If none of this works, the algorithm is stopped (and possibly restarted from another initial value). After all, the point of Case c can be a local optimum in contrived examples. Also see the discussion of the convergence issue at the end of this section.

B. Computational Cost

The computationally most expensive steps of the algorithm are the evaluations of the spectral radius of X and the corresponding right and left eigenvector. These do not only occur in line four of the algorithm, but mainly in line six during the line search. (Actually, except for the first step in the iteration, nothing needs to be calculated in line four, since ρ, p, and v were calculated already during the last execution of line six). In order to decrease this cost, the following things can be done.

1) The number of eigenvalue/eigenvector calculations is kept low, using an efficient line search algorithm by Fletcher [10, pp. 33]. This algorithm finds a point, satisfying the (modified) Wolfe–Powell conditions (normally given for minima instead of maxima)

$$\Phi(t) \geq \Phi(0) + t\rho_F \frac{d\Phi}{dt}(0)$$
$$\left|\frac{d\Phi}{dt}(t)\right| \leq \sigma_F \frac{d\Phi}{dt}(0) \tag{6}$$

where ρ_F and $\sigma_F \geq \rho_F$ are fixed parameters. Typically $\sigma_F = 0.5$ and $\rho_F = 0.01$, for a moderately accurate line search [10, pp. 30]. Note that Fletcher's algorithms are designed for smooth optimization problems. However, because the local maximum does not occur at a nondifferentiable point, normally no problems arise. (However, as we have not proved that no nondifferentiable maxima of Case c can exist, to be really sure one should use a line search algorithm that can also take into account this case).
2) For the calculation of ρ, p, and v, some profit can be gained from the fact that only the principal eigenvalue and the corresponding eigenvectors are needed and the fact that previous calculations for nearby X provide good initial estimations. Both the well-known power algorithm (not to be confused with the PIA for the structured singular value, called power algorithm for its resemblance with the classical power algorithm) and the inverse iteration algorithm can save numerous computations. However, the power algorithm is not efficient if the maximal eigenvalue of X is not well separated (in absolute value) from the other eigenvalues, and the inverse iteration algorithm can converge to the wrong eigenvector with a bad initial guess for the eigenvalue. Therefore, the price of the computational improvement is a number of heuristic rules to handle the distinction between different cases or the loss of the ascent property. However, our experience shows that problems are rare, and efficiency can be increased when the line search is made more robust against inaccurate gradient information. In most numerical experiments, however, we have used the MATLAB function "eig" [11], to calculate a full eigenvalue decomposition.

C. Remarks on the Convergence Issue

Finally, we come back to the convergence issue, already mentioned in Section I. For discrete-time ascent algorithms, convergence is a subtle issue. The ascent property, together with the boundedness of Q, proves convergence of ρ (or Φ), not of Q. Whether Q converges can depend on properties of the line search (although the fact that $\mathcal{Q}$ is compact already avoids many problems). However, since we are only interested in ρ (to estimate μ), this is no problem. For this reason we do not put more energy in proving convergence results for Q. Nevertheless, it is important that not too many points are attractive, especially if they have low ρ. Saddle points can be attractive in theory, but only from special initial conditions. Moreover, random perturbations are often sufficient to escape. Because the function is nondifferentiable, in theory also nondifferentiable points could be attractive, without being locally maximal. However, we have not been able to construct such examples (also see [5]).

In summary, the algorithm keeps increasing ρ, and not in a trivial way, like taking directions that converge to a direction of no change, or taking step sizes converging to zero where it could easily do better.

IV. The Conjugate Gradient Algorithm

In this section we derive the CGA on the manifold $\mathcal{Q}$, solving the optimization problem as given by Definition 1. For the CGA the search direction at a point Q_k is determined in most steps by the gradient at Q_k and the previous search direction at Q_{k-1}. In most versions, the gradient direction itself is taken every nth step (where n is the problem dimension). Smith proposes the following formula for the search direction $H_k Q$, at a point Q_k using the gradients $G_k Q$ and $G_{k-1}Q$ and the previous search direction $H_{k-1}Q$. It is very similar to the Polak–Ribière formula for the CGA in Euclidean spaces, but tangent vectors at the point Q_{k-1} from the previous iteration step are adapted to Q_k by parallel translation along the geodesic through Q_{k-1} and Q_k

$$H_k = G_k + \gamma_k \tau H_{k-1}, \qquad \text{where } \gamma_k = \frac{\langle G_k - \tau G_{k-1}, G_k\rangle}{\langle G_{k-1}, H_{k-1}\rangle}$$

where τ denotes parallel translation from Q_{k-1} to Q_k, applied to matrices S representing a tangent vector SQ at Q. It follows from Theorem 1 that $\tau H_{k-1} = H_{k-1}$. To calculate τG_{k-1}, (4) is needed. This leads to the following algorithm.

Algorithm 2 (CGA):

1. initialize Q_0
2. $k = 0$
3. **repeat**
4. compute $\rho = \rho(Q_k M)$ and corresponding right and left eigenvectors p and v, such that $v^H p = 1$
5. $G_k = \Phi(Q)\pi(vp^H - pv^H)$
6. if $(k \bmod n) = 0$ then $H_k = G_k$
7. else $\tau G_{k-1} = \exp\left(\frac{1}{2}H_{k-1}t\right) G_{k-1} \exp\left(-\frac{1}{2}H_{k-1}t\right)$
8. $\gamma = \frac{\langle G_k - \tau G_{k-1}, G_k\rangle}{\langle G_{k-1}, H_{k-1}\rangle}$
9. $H_k = G_k + \gamma H_{k-1}$
10. $t = \arg\max_{t>0}\ \rho(\exp(H_k t)Q_k M)$
11. $Q_{k+1} = \exp(H_k t)Q_k$
12. $k = k + 1$
13. **until** convergence.

Remark 4: For the calculation of $\exp(Ht)$ for different values of t, the same remark holds as for the SAA (Remark 2).

Remark 5: For the line search of line ten, the same strategy as for the SAA is used. However, since the performance of the CGA is more sensitive to the quality of this line search, the conditions in (6) are made stronger by taking $\sigma_F = 0.1$ ($\rho_F = 0.01$) [10, p. 85].

The CGA is known to have better asymptotic convergence properties than the SAA (assuming convergence to a differentiable point). The additional cost of computing the conjugate gradient search direction is small compared to the cost of the function evaluations and gradient calculations, especially when the structured matrices consist of small blocks, since these computations can be performed on the block components separately. However, due to the stronger requirements for the line search (Remark 5), individual iterations may require more computations. In Section V we show that for the present problem, both algorithms are more or less equivalent.

V. Numerical Experiments

In this section we support our results with numerical experiments. We give an idea of the efficiency of our algorithms and compare them with the PIA as implemented in the MATLAB μ

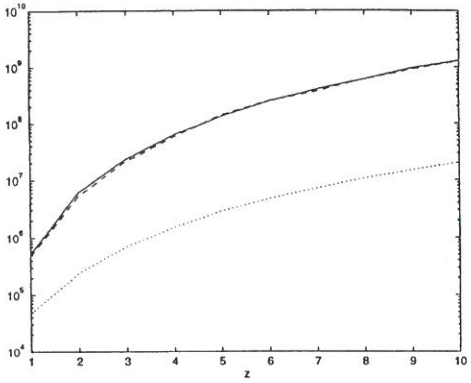

Fig. 1. Number of floating point operations for CGA (full line), SAA (dashed line), and PIA (dotted line), for matrices with different sizes and a block structure, consisting of two repeated scalar blocks of size z and two full blocks of size z, where $z = 1, \cdots, 10$.

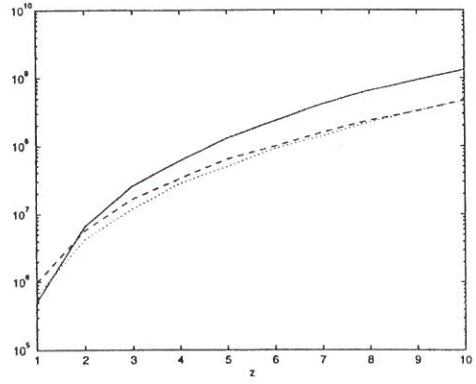

Fig. 2. Number of floating point operations for SAA, with eigenvector/eigenvalue computation with the MATLAB function "eig" (full line), with power iterations (dashed line), and with inverse power iterations (dotted line).

toolbox [12]. The MATLAB code for our algorithms is available via anonymous ftp at ftp.esat.kuleuven.ac.be in the directory /pub/SISTA/dehaene/prog/ccmu. All experiments were performed with the same convergence test. The algorithm stops if the relative increase of Φ drops below $1E-4$ for two consecutive steps.

Fig. 1 compares the computational cost of SAA, CGA, and PIA, for complex matrices with different sizes and a similar block structure, consisting of two repeated scalar blocks of size z and two full blocks of size z, where $z = 1, \cdots, 10$. The matrices were generated at random and for each size the computational cost was averaged over 100 experiments with different matrices. Experiments with different block structures gave no significant difference.

The SAA and CGA require approximately the same amount of computations. (With weaker convergence criteria the SAA tends to take less time than the CGA.) As mentioned above, the PIA takes considerably less time. However, it should be noted that we are comparing an experimental version of the SAA and CGA with a fine-tuned version of PIA. Similar results were obtained with different convergence criteria. The experiments of Fig. 1 used random initial values. The effect of using different initial values is small.

Fig. 2 shows the computational cost for the CGA (similar results are obtained with the SAA), when the eigenvectors and eigenvalues are computed with power iterations or inverse iterations (see Section III). For both power iterations and inverse iterations classical convergence tests [13] were used with a tolerance of $1E-4$. For power iterations an upper limit of 50 iterations was set. For inverse iterations this limit was set to ten iterations. If the limit is reached the eigenvectors and eigenvalues are calculated with the MATLAB function "eig." Fig. 2 shows the result for matrices of different sizes, generated in the same way as for the comparison of the SAA, CGA, and PIA, described above. Note that power iterations and inverse iterations require approximately the same number of operations.

VI. Conclusion

We have derived new algorithms for computing the structured singular value. The algorithms implement ascent methods on a manifold of structured unitary matrices. Unlike the existing PIA algorithm, the new algorithms always find a nontrivial lower-bound. However, they are less efficient than the PIA algorithm, and we propose to combine both algorithms. We also hope that a combination of our analysis of the problem with the ideas behind the PIA algorithm could lead to better insight in the convergence behavior of the PIA algorithm and possibly to new algorithms with the efficiency of the PIA algorithm and guaranteed convergence.

References

[1] J. C. Doyle, "Analysis of feedback systems with structured uncertainties," *Proc. Inst. Elec. Eng.*, vol. 129, pp. 242–250, 1982.
[2] A. Packard, M. K. H. Fan, and J. Doyle, "A power method for the structured singular value," in *Proc. 27th Conf. Decision Contr.*, Austin, TX, Dec. 1988, pp. 2132–2137.
[3] C. Yi, "Robustness analysis and controller design for systems with structured uncertainties," Ph.D. dissertation, K. U. Leuven, May 1995.
[4] C. Yi and B. DeMoor, "Continuous time power iteration for computing a lower bound of μ," in *Proc. Amer. Contr. Conf.*, Baltimore, MD, June 29–July 1, 1994, pp. 1416–1417.
[5] J. Dehaene, "Continuous-Time matrix algorithms, systolic algorithms and adaptive neural networks," Ph.D. dissertation, K. U. Leuven, Oct. 1995; also ftp.esat.kuleuven.ac.be/pub/SISTA/dehaene/-phd/.
[6] S. T. Smith, "Geometric optimization methods for adaptive filtering," Ph.D. dissertation, Harvard Univ., Cambridge, MA, May 1993.
[7] D. H. Sattinger and O. L. Weaver, *Lie Groups and Algebras with Applications to Physics, Geometry and Mechanics*, vol. 61 of Applied Mathematical Sci. New York: Springer-Verlag, 1993.
[8] S. Helgason, *Differential Geometry, Lie Groups and Symmetric Spaces*. New York: Academic, 1978.
[9] T. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*. New York: Springer-Verlag, 1982.
[10] R. Fletcher, *Practical Methods of Optimization*. New York: Wiley, 1993.
[11] The Math Works Inc., Natick, MA 01760 USA. *MATLAB Reference Guide*, 1992.
[12] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith, *μ-Analysis and Synthesis Toolbox*. Natick, MA: The MathWorks, 1993.
[13] G. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1991.

A Counterexample of "Comments on 'Stability Margin Evaluation for Uncertain Linear Systems'"

Yong-Yan Cao and You-Yian Sun

***Abstract*— In this paper, a counterexample of the above-mentioned paper[1] is presented, and it shows that the main result of this paper is not correct.**

Consider the uncertain linear dynamic system of order n

$$\dot{x}(t) = A(t)x(t) = [A_0 + \Delta A(t)]x(t) \tag{1}$$

where A_0 is the nominal stable system matrix and $\Delta A(t)$ is the time-varying uncertainty.

Recently, Gong and Thompson [1] have given a stability margin evaluation method for this unstructured matrix. The criterion in Theorem 2 of this paper is

$$\Delta A^T \Delta A < \tfrac{1}{4}\sigma^2_{\min}(U + U^T)A_0^T A_0.$$

It is also claimed in this paper that this bound is the tightest bound possible for all unstructured perturbations, such that (1) keeps its asymptotic stability. But this bound was shown not to be the tightest for all unstructured perturbations by the above-mentioned paper,[1] and a new bound also has been given; however, it is not right.

Let us consider the example of the paper[1] whose system matrix is

$$A_0 = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}.$$

In this paper, it is shown that the system is guaranteed to be asymptotically stable by Theorem 1 if

$$\Delta A^T \Delta A < \frac{0.707^2}{0.708^2} A_0^T A_0. \tag{2}$$

Let

$$\Delta A = \begin{bmatrix} 3.1 & 1.9 \\ 0 & 0 \end{bmatrix}.$$

It is obvious that $\Delta A^T \Delta A < (0.707^2/0.708^2)A_0^T A_0$. But

$$A_0 + \Delta A = \begin{bmatrix} 0.1 & -0.1 \\ 1 & 0 \end{bmatrix}$$

is not stable because its two eigenvalues are $0.05 \pm 0.3122i$.

References

[1] C. Gong and S. Thompson, "Stability margin evaluation for uncertain linear systems," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 548–550, 1994.

Manuscript received April 17, 1996.
The authors are with the Institute of Industrial Process Control, Zhejiang University, Hangzhou, 310027, P.R. China (e-mail: yycao@iipc.zju.edu.cn).
Publisher Item Identifier S 0018-9286(97)07640-X.

[1] J.-H. Su, *IEEE Trans. Automat. Contr.*, vol. 39, pp. 2523–2524, 1994.

Filtering Systems with Finite-Dimensional Estimation Algebras

Rui-Tao Dong, Wing Shing Wong, and Stephen S.-T. Yau

***Abstract*— Estimation algebra turns out to be a useful concept in the investigation of finite-dimensional nonlinear filters. In this paper we study the natural question of classifying all filtering systems with finite-dimensional estimation algebras up to state-space diffeomorphism. In particular, we present some results on partial differential equations arising from the study of stochastic systems and nonlinear filtering problems.**

***Index Terms*—Estimation algebra, nonlinear filters, under-determined partial differential equation.**

I. Introduction

In many filtering systems and stochastic control problems, one has to deal with elliptic differential operators of a certain type. For example, consider a filtering problem based on the following signal observation model:

$$\begin{aligned} dx(t) &= f(x(t))\,dt + dv(t), \qquad x(0) = x_0 \\ dy(t) &= Hx(t)\,dt + dw(t), \qquad y(0) = 0 \end{aligned} \tag{1}$$

in which x, v, y, and w, are respectively, R^n, R^p, R^m, and R^m-valued processes, and v and w have components that are independent standard Brownian processes.

$\rho(t, x)$, the conditional probability density of the state $x(t)$, given the observation $\{y(s): 0 \le s \le t\}$, is determined by the Duncan–Mortensen–Zakai equation, which in the *un-normalized* form is given by (see [9] for example)

$$\begin{aligned} \frac{d}{dt}\sigma(t,x) &= L_0\sigma(t,x)\,dt + \sum_{i=1}^{m} L_i\sigma(t,x)\,dy_i(t) \\ \sigma(0,x) &= \sigma_0 \end{aligned} \tag{2}$$

where

$$L_0 = \frac{1}{2}\sum_{i=1}^{n}\frac{\partial^2}{\partial x_i^2} - \sum_{i=1}^{n} f_i\frac{\partial}{\partial x_i} - \sum_{i=1}^{n}\frac{\partial f_i}{\partial x_i} - \frac{1}{2}\sum_{i=1}^{m} h_i^2$$

and for $i = 1, \cdots, m$, L_i is the zero-degree differential operator of multiplication by h_i. (If p is a vector, we use the notation p_i to represent the ith component of p.) σ_0 is the probability density of the initial point x_0. When the observation is absent, that is $h = 0$, then (2) is simply the Kolgomorov equation.

It is important to find efficient ways to solve (2), which is the subject of many research studies in nonlinear filtering theory. A particularly useful concept is that of an estimation algebra, which was introduced in [3], [4], and [12]. The survey paper [11] provides a good introduction and many useful references to the concept. It is defined to be the Lie algebra of differential operators generated

Manuscript received July 17, 1995; revised July 13, 1996. This work was supported in part by the Army under Grant DAAH0493G0006.
R. T. Dong is with the Department of Mathematics, Johns Hopkins University, Baltimore, MD 21218 USA.
W. S. Wong is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T. Hong Kong.
S. S.-T. Yau is with the Laboratory of Control and Information, Department of Mathematics, Statistics & Computer Sciences, University of Illinois at Chicago, Chicago, IL 60607-7045 USA.
Publisher Item Identifier S 0018-9286(97)07641-1.