TR-86-8

AN ALGORITHM TO COMPUTE THE STRUCTURED SINGULAR VALUE

by

Michael K.H. Fan

An Algorithm to Compute the Structured Singular Value

by

Michael K.H. Fan

System Research Center
University of Maryland
College Park, MD 20742

Abstract

The concept of *structured singular value* was recently introduced by Doyle as a tool for analysis and synthesis of feedback systems with structured uncertainties. It is a key to the design of control systems under joint robustness and performance specifications and it nicely complements the H^{∞} approach to control system design. This report proposes an algorithm to compute the structured singular value.

1. Introduction and preliminaries

The concept of *structured singular value* was recently introduced by Doyle [1] as a tool for analysis and synthesis of feedback systems with structured uncertainties. It is a key to the design of control systems under joint robustness and performance specifications and it nicely complements the H^{∞} approach to control system design. [2]

Throughout the note, given any square complex matrix M, we denote by $\rho(M)$ its spectrum radius, by $\overline{\sigma}(M)$ its largest singular value and by M^H its complex conjugate transpose. Given any complex vector x, x^H indicates its complex conjugate transpose and $\|x\|$ its Euclidean norm. We also make use of the following notation and nomenclature, largely inspired from that used in. [1] We will call *block-structure* of size m any m-tuple $\mathcal{k} = (k_1, \cdots, k_m)$ of positive integers. Given a block-structure $\mathcal{k}$ of size m, we will make use of the family of diagonal matrices

$$\mathcal{d} = \{\text{block diag}(d_1 I_{k_1}, \cdots, d_m I_{k_m}) \mid d_i \in \mathbb{R}\} \; ; \tag{1.1}$$

and, for any positive scalar δ (possibly ∞), of the family of block diagonal matrices

$$X_\delta = \{\text{block diag}(\Delta_1, \cdots, \Delta_m) \mid \Delta_i \text{ is a } k_i \times k_i \text{ complex matrix satisfying } \overline{\sigma}(\Delta_i) \leq \delta\} \tag{1.2}$$

All of the above have dimension $n \times n$, where

$$n = \sum_{j=1}^{m} k_j \; . \tag{1.3}$$

The following definition corresponds to the case of "no repeated blocks" in. [1]

Definition 1.1

The *structured singular value* $\mu(M)$ of a complex $n \times n$ matrix M ***with respect to block-structure*** $\mathcal{k}$ is the positive number μ having the property that

$$\det(I + M\Delta) \neq 0 \quad \text{for all } \Delta \in X_\delta \tag{1.4}$$

if, and only if,

$$\delta\mu < 1 \; . \tag{1.5}$$

In other words, $\mu(M)$ is 0 if there is no Δ in X_∞ such that $\det(I + M\Delta) = 0$, and $\left(\min_{\Delta \in X_\infty} \{ \overline{\sigma}(\Delta) \mid \det(I + M\Delta) = 0 \} \right)^{-1}$ otherwise.

[]

It should be noted that $\mathcal{d}$, X_δ, and $\mu(M)$ all depend on the underlying block-structure. In most instances, we will not explicitly specify this block-structure.

We will make repeated use of the following easily derived fact. [1]

Fact 1.0 For all $D \in \mathcal{d}$,

$$\mu(M) = \mu(e^{D} M e^{-D}) \tag{1.6}$$

In order to evaluate the structured singular value, more manageable expressions than those provided in Definition 1.1 are desirable. Such expressions are provided by the following fact. [1]

Fact 1.1 For block-structures of size less than 4,

$$\mu(M) = \inf_{D \in \mathcal{d}} \overline{\sigma}(e^{D} M e^{-D}) \; . \tag{1.7}$$

[]

In fact, in many (but not all) cases, (1.7) is correct with block-structures of larger size. A counterexample, due to Doyle, [3] which shows that (1.7) is violated, is given in Appendix A.

In, [1] Doyle proposed an algorithm, which is essentially based upon first derivatives, to solve problem (1.7). Since when the largest singular value of $e^D Me^{-D}$ is simple, the square of $\overline{\sigma}(e^D Me^{-D})$ is continuously differentiable in D. Hence it is possible to express the first and second derivatives analytically such that, locally, Newton's method could be applied to solve problem (1.7). This report proposes a modified algorithm to compute the structured singular value based upon the first and second derivatives. In section 2, we will discuss a first order algorithm which mostly follows the line in. [1] In section 3, we will discuss the continuity properties of hermitian matrices. Finally, in section 4, a second order algorithm is presented.

2. A first order algorithm

In this section, we will discuss algorithms to solve the right hand side of (1.7) or, equivalently,

$$\inf_{D \in d} \|e^{D} M e^{-D}\|^2 \tag{2.1}$$

by means of (generalized) gradient search method. Since $\|e^{D} M e^{-D}\|^2$ is convex [4] in D, it results that all stationary points are global minima. Since for any $\alpha \in \mathbb{R}$, $e^{D} M e^{-D} = e^{\alpha I + D} M e^{-\alpha I - D}$, without loss of generality, we assume that $d_m = 0$. Recall that $D = \text{blockdiag}\,\{d_1 I_{k_1}, \cdots, d_m I_{k_m}\}$. Define $\underline{d} = [d_1 \cdots d_{m-1}]^T$, $g(\underline{d}) = \|e^{D} M e^{-D}\|^2$ and $H(\underline{d}) = (e^{D} M e^{-D})$. Note that $g(\underline{d})$ is continuous but not always differentiable. However the following property holds

Proposition 2.1 For any $\underline{h}$, the following expression exists

$$\lim_{t \to 0^+} \frac{g(t\underline{h}) - g(0)}{t} .$$

[]

Definition 2.1 $\underline{h}$ is said to be a *descent direction* for $g(\underline{d})$ at $\underline{d} = 0$ if there exists a $\delta > 0$ such that for every $t \in (0, \delta)$

$$g(t\underline{h}) < g(0) .$$

[]

Definition 2.2 A unit norm vector $\underline{h}$ is said to be a *steepest descent direction* for $g(\underline{d})$ at $\underline{d} = 0$ if $\underline{h}$ is a descent direction and a solution of

$$\min_{\underline{h}} \left\{ \lim_{t \to 0^+} \frac{g(t\underline{h}) - g(0)}{t} \mid \|\underline{h}\| = 1 \right\} .$$

[]

Suppose that $H(0)$ has a simple largest eigenvalue λ_1, we denote v_1 the unit norm eigenvector corresponding to λ_1. Thus gradient of $g(\underline{d})$ at $\underline{d} = 0$ can be computed component-wise as follows. For $j = 1, \cdots, m-1$,

$$\nabla g_j(0) = v_1^H H_j v_1 \tag{2.2}$$

where $H_j = 2Re\,(H(0)^H \frac{\partial H(0)}{\partial d_j})$. So $-\nabla g(0)/\|\nabla g(0)\|$ is a (steepest) descent direction for $g(\underline{d})$ at $\underline{d} = 0$.

Proposition 2.2 If λ_1 is simple and $\nabla g(0) = 0$, then $\mu(M) = \overline{\sigma}(M)$.

Proof. See. [1, 4]

[]

Corollary 2.1 If (2.1) is achievable and the corresponding largest singular value is simple, then (1.7) holds (m needs not to be less than four).

[]

In the case that the largest eigenvalue of $H(0)$ has multiplicity q, $q > 1$, $g(\underline{d})$ is then not continuously differentiable at $\underline{d} = 0$. Therefore the gradient is not well defined. In order to find a descent direction for $g(\underline{d})$, a generalized gradient is introduced. Let P_1 denote the set containing all the unit norm eigenvectors corresponding to λ_1. Define

$$\nabla_2 = \{\underline{y} = (y_1, \cdots, y_{m-1}) \mid y_j = \underline{x}^H H_j \underline{x}, \ \underline{x} \in P_1\} \ . \tag{2.3}$$

Clearly, if λ_1 is simple, ∇_2 reduces to $\{\nabla g(0)\}$.

Proposition 2.3 When $m \leq 3$, ∇_2 is convex.

Proof. See. [1]

[]

Proposition 2.4 $\mu(M) = \overline{\sigma}(M)$ if and only if $0 \in \nabla_2$.

Proof. See. [1]

[]

Proposition 2.5 Suppose $0 \notin \mathrm{co}\nabla_2$ and vector $\underline{h}$ has the property that

$$< \underline{h}, \underline{y} > \; < 0 \quad \text{for all } \underline{y} \in \nabla_2 \ , \tag{2.4}$$

then $\underline{h}$ is a descent direction of $g(\underline{d})$ at $\underline{d} = 0$, where $\mathrm{co}\nabla_2$ denotes the convex hull of ∇_2.

Proof. See. [1]

[]

Corollary 2.2 Assume that (2.1) is achievable and $m \leq 3$, then (1.7) holds.

Proof. Assume that D^* solves (2.1). By Proposition 2.5, we have $0 \in \mathrm{co}\nabla_2$ where ∇_2 is defined in terms of $e^{D^*} M e^{-D^*}$. By Proposition 2.3, since ∇_2 is convex, $0 \in \nabla_2$. Finally, by Proposition 2.4, since $0 \in \nabla_2$, we conclude that

$$\mu(M) = \mu(e^{D^*} M e^{-D^*}) = \overline{\sigma}(e^{D^*} M e^{-D^*}) = \inf_{D \in d} \overline{\sigma}(e^{D} M e^{-D}) \ . \tag{2.5}$$

[]

Proposition 2.6 Let $\underline{h} = -Nr(\mathrm{co}\nabla_2)$, then $\underline{h}/\|\underline{h}\|$ is a steepest descent direction of $g(\underline{d})$ at $\underline{d} = 0$, where $Nr(\mathrm{co}\nabla_2)$ denotes the nearest point to the origin in $\mathrm{co}\nabla_2$.

[]

We now are ready to state a first order algorithm for computing (2.1).

Algorithm 2.1

Step 1.

Data $M_0 = M$, $D_0 = 0$ ($\underline{d}_0 = 0$).

$k = 0$.

Step 2.

Set $M_{k+1} = e^{D_k} M_k e^{-D_k}$.

Define search direction $\underline{h}$ to be $-Nr(\mathrm{co}\nabla_2)$ where ∇_2 is defined in terms of M_{k+1}.

Step 3.

Perform line search to find the step size α.

Step 4.

$\underline{d}_{k+1} = \underline{d}_k + \alpha \underline{h}$ (D_{k+1} is therefore updated).

Set $k = k+1$, go to step 2.

[]

Proposition 2.7 Let $D^* = \sum_{k=1}^{\infty} D_k$ where $\{D_k\}$ is the sequence generated by Algorithm 2.1. Then

$$\overline{\sigma}(e^{D^*} M e^{-D^*}) = \inf_{D \in d} \overline{\sigma}(e^{D} M e^{-D}) \ .$$

[]

Let $\{v_1, \cdots, v_q\}$ be a basis for P_1. (recall that P_1 denotes the set containing all the unite norm eigenvector corresponding to λ_0) Define $V = [v_1, \cdots, v_q]$, $\overline{H}_j = V^H H_j V$ and $P_2 = \{\underline{x} \in C^q \mid \underline{x}^H \underline{x} = 1\}$. Note that $\overline{H}_j$ is of size $q \times q$. By using these notation, ∇_2 could be expressed in a more manageable way as follows

$$\nabla_2 = \{ f\,(\underline{x}) \mid f^{\,j}(\underline{x}) = \underline{x}^H \overline{H}_j \underline{x}\,,\;\; j = 1, \cdots, m-1,\;\; \underline{x} \in P_2 \} \tag{2.6}$$

The following algorithm, [1] which is based upon (2.6), is to find $Nr\,(\mathrm{co}\nabla_2)$.

Algorithm 2.2

Step 1.

Pick any $\underline{v}_0 \in P_2$ and let $\underline{x}_0 = f\,(\underline{v}_0)$. Set $k = 0$.

Step 2.

Set $\underline{x}_{k+1} = Nr\,(\mathrm{co}[\underline{x}_k, f\,(\underline{v}_k)])$.

Step 3.

Let $\underline{v}_{k+1}$ be any unit vector for $\lambda_{\min}(\sum_{j=1}^{m-1} \underline{x}_{k+1}^{j} \overline{H}_j)$, where $\lambda_{\min}$ denotes the smallest eigenvalue.

Step 4.

Set $k = k+1$, go to step 2.

[]

Proposition 2.8 Let $\{\underline{x}_k\}$ be the sequence generated by Algorithm 2.2, then $\{\underline{x}_k\}$ converges to $Nr\,(\mathrm{co}\nabla_2)$. Furthermore, let $\underline{x}^*$ denote the limit and suppose that $\underline{x}^* \neq 0$, then $\sum_{j=1}^{m-1} \underline{x}^{*j} \overline{H}_j$ is a strictly positive definite matrix.

Proof. It is shown in [1] that any convergent subsequence of $\{\underline{x}_k\}$ converges to $Nr\,(\mathrm{co}\nabla_2)$. Since the sequence $\{\|\underline{x}_k\|\}$ is bounded and $Nr\,(\mathrm{co}\nabla_2)$ has a unique solution, it is true that $\{\underline{x}_k\}$ itself converges and the limit is $Nr\,(\mathrm{co}\nabla_2)$. Furthermore, let $\underline{v}^*$ be any accumulation point of the sequence $\{\underline{v}_k\}$. By the algorithm and the definition of limit, we have $<\underline{x}^*, f\,(\underline{v}^*)> = \lambda_{\min}(\sum_{j=1}^{m-1} \underline{x}^{*j} \overline{H}_j)$. If $\lambda_{\min}(\sum_{j=1}^{m-1} \underline{x}^{*j} \overline{H}_j)$ is not greater than zero, we have $\underline{x}^* \neq Nr(\text{co}[\underline{x}^*, f(\underline{v}^*)])$ which leads to a contradiction.

[]

As mentioned above, when λ_1, the largest eigenvalue of $H(0)$, is simple, i.e. $q=1$, ∇_2 reduces to $\{\nabla g(0)\}$ and, therefore, $Nr(\text{co}\nabla_2) = \nabla g(0)$. In the case that $q=2$, it can be shown that the boundary of $\text{co}\nabla_2$ is a second order curve (or surface), possibly degenerate, in $\mathbb{R}^{m-1}$. Hence, $Nr(\text{co}\nabla_2)$ could also be solved analytically. Based on this observation, for any q, we will propose another algorithm to compute $Nr(\text{co}\nabla_2)$. Now we proceed this by giving more details about the case $q=2$. For $j=1, \cdots, m-1$, let

$$\overline{H}_j = \begin{bmatrix} a_j & b_j \\ b_j^H & c_j \end{bmatrix} \tag{2.7}$$

where $a_j, c_j \in \mathbb{R}$ and $b_j \in C$. Define ∇_2 accordingly. Also define $\underline{l}$ to be the vector in $\mathbb{R}^{m-1}$ such that the jth component of $\underline{l}$ is $(a_j + c_j)/2$, and, define A to be the matrix in $\mathbb{R}^{(m-1)\times 3}$ with the jth row being $[(a_j - c_j)/2 \;\; \text{Re}(b_j) \;\; \text{Im}(b_j)]$. Recall that, for $q=2$, $P_2 = \{\underline{x} \in C^2 \mid \underline{x}^H \underline{x} = 1\}$. Let $S = \{\underline{x} \in \mathbb{R}^3 \mid \underline{x}^T \underline{x} = 1\}$. We define $g(\underline{x})$ to be an affine function such that $g(\underline{x}) = A\underline{x} + \underline{l}$.

Proposition 2.9 $\nabla_2 = g(S)$.

Proof. See. [1]

By the Proposition 2.9, it becomes possible to image how the set ∇_2 looks like for the case $q=2$ and, fortunately, in this case the boundary of $\text{co}\nabla_2$ is either a point, an interval, an ellipse in $\mathbb{R}^2$ or an ellipsoid in $\mathbb{R}^3$. Therefore, finding the nearest point to the origin in $\text{co}\nabla_2$ is straightforward. Perform the singular value decomposition of A such that then $Nr(\text{co}\nabla_2) = U_A \; Nr(Q_2 + U_A^T \underline{l})$.

[]

We now state another algorithm to compute $Nr(\text{co}\nabla_2)$.

Algorithm 2.3

Step 1.

Pick any $\underline{v}_0 \in P_2$ and let $\underline{x}_0 = f(\underline{v}_0)$. Set $k = 0$.

Step 2.

Let $\underline{u}_k$ be any unit vector for $\lambda_{\min}(\sum_{j=1}^{m-1} \underline{x}_k^j \overline{H}_j)$.

Step 3.

Define

$$\overline{\overline{H}}_j = \begin{bmatrix} \underline{v}_k^H \\ \underline{u}_k^H \end{bmatrix} \overline{H}_j \begin{bmatrix} \underline{v}_k & \underline{u}_k \end{bmatrix} \quad j=1, \cdots, m-1 \; .$$

Define $\overline{f}$ in terms of $\overline{\overline{H}}_j$. Analytically find the solution $\underline{w}$ such that $\overline{f}(\underline{w})$ is the nearest point to the origin in set P_2. Set $\underline{v}_{k+1} = [\underline{v}_k \;\; \underline{u}_k]\underline{w}$ and $\underline{x}_{x+1} = f(\underline{v}_{k+1})$.

Step 4.

Set $k = k+1$, go to step 2.

[]

Proposition 2.11 Proposition 2.8 also holds for Algorithm 2.3.

[]

Let

$$A = U_A \Sigma_A V_A^T \tag{2.8}$$

where U_A and V_A are orthogonal matrices in $\mathbb{R}^{(m-1)\times(m-1)}$ and $\mathbb{R}^{3\times 3}$ respectively, and

$$\Sigma_A = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ \cdot & 0 & \sigma_3 \\ \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot \\ 0 & 0 & 0 \end{bmatrix} \tag{2.9}$$

provided that Σ_A has appropriate dimension. The following Proposition gives the solutions for all cases in terms of the rank of A .

Proposition 2.10

case 1. $\text{rank}(A) = 0$

$Nr\,(\text{co}\nabla_2) = \underline{l}$.

case 2. $\text{rank}(A) = 1$

$Nr\,(\text{co}\nabla_2) = U_A \;\; Nr\,(\text{co}[(\sigma_1,0,\cdots,0)^T + U_A^T\underline{l},\; (-\sigma_1,0,\cdots,0)^T + U_A^T\underline{l}])$.

case 3. $\text{rank}(A) = 2$

Let Q_1 denote the set

$$\{\underline{x} = (x_1,x_2,0,\cdots,0) \mid \underline{x} \in \mathbb{R}^{m-1},\; \frac{x_1^2}{\sigma_1^2}+\frac{x_2^2}{\sigma_2^2} \leq 1\} \;,$$

then $Nr\,(\text{co}\nabla_2) = U_A \;\; Nr\,(Q_1 + U_A^T\underline{l})$.

case 4. $\text{rank}(A) = 3$

Let Q_2 denote the set

$$\{\underline{x} = (x_1,x_2,x_3,0,\cdots,0) \mid \underline{x} \in \mathbb{R}^{m-1},\; \frac{x_1^2}{\sigma_1^2}+\frac{x_2^2}{\sigma_2^2}+\frac{x_3^2}{\sigma_3^2} \leq 1\} \;,$$

$$g_1(\underline{d}) = \lambda_{\max}([u_1, \cdots, u_q]^H D [u_1, \cdots, u_q] - [v_1, \cdots, v_q]^H D [v_1, \cdots, v_q]) \quad (2.10)$$

where $\lambda_{\max}(A)$ denotes the largest eigenvalue of A, and q is the multiplicity of $H(0)$.

Proposition 2.12 If $g_1(\underline{h})<0$, then $\underline{h}$ is a descent direction. If $g_1(\underline{h})>0$, then $\underline{h}$ is not a descent direction.

[]

Proposition 2.13 The following three statements are equivalent.

1. $\underline{h}$ is a steepest descent direction of $g(\underline{d})$ at $\underline{d}=0$.

2. $\underline{h} = \frac{\underline{h}_1}{\|\underline{h}_1\|}$ where $\underline{h}_1 = -Nr(\mathrm{co}\nabla_2)$.

3. $\underline{h}$ is a unit norm descent direction and $\underline{h}$ solves

$$\min_{\underline{d}}\{g_1(\underline{d}) \mid \|\underline{d}\| = 1\} \quad . \quad (2.11)$$

[]

3. Continuity properties of hermitian matrix

Suppose $H(t)$ is hermitian and real analytic in t, it is well-known that by appropriate ordering of eigenvalues $\{\lambda_i\}$ and selection of eigenvectors $\{v_i\}$, it is possible to pair eigenvalues and eigenvectors $\{\lambda_i(t), v_i(t)\}$, such that both $\lambda_i(t)$ and $v_i(t)$ are analytic in t and $H(t)v_i(t) = \lambda_i(t)v_i(t)$ for all t and i. At values of t, where $H(t)$ has simple eigenvalues this is trivial. At degenerate points, it require selection of the λ_i and v_i such that analyticity is retained through (isolated) point where eigenvalues coalesce. Note that with this ordering, the $\{\lambda_i\}$ are not necessarily linearly ordered. In this section, we will explore the continuity properties of hermitian matrix such that by using these properties, we could derive a second order algorithm in the next section for computing (2.1).

Let $H(t)$ be an $n \times n$ hermitian matrix and suppose that it is real analytic in t. We denote $\lambda_1(t)$ the eigenvalue corresponding to the spectral radius of $H(t)$ and $\underline{v}_1(t)$ the corresponding eigenvector such that,

$$H(t)\underline{v}_1(t) = \lambda_1(t)\underline{v}_1(t) \tag{3.1}$$

We assume that $\lambda_1 = \lambda_1(0)$ is simple. Since $H(t)$, $\lambda_1(t)$ and $\underline{v}_1(t)$ are analytic, we could express them in the form of Taylor series of t at $t=0$ as follows

$$H(t) = H_0 + t\dot{H} + \frac{1}{2}t^2\ddot{H} + o(t^2) \ , \tag{3.2}$$

$$\lambda_1(t) = \lambda_1 + t\dot{\lambda}_1 + \frac{1}{2}t^2\ddot{\lambda}_1 + o(t^2) \tag{3.3}$$

and

$$\underline{v}_1(t) = \underline{v}_1 + tV_{\perp}\underline{x} + \frac{1}{2}t^2\underline{y} + o(t^2) \ . \tag{3.4}$$

where $\underline{x} \in C^{n-1}$, $\underline{y} \in C^n$ and $[\underline{v}_1 \mid V_{\perp}]$ is a unitary matrix such that

$$H_0 V_\perp = V_\perp \Lambda_\perp \ . \tag{3.5}$$

where $\Lambda_\perp$ is a diagonal matrix with all eigenvalues of H_0 except λ_1 in the diagonal. Therefore we have

$$(H_0 + t\dot{H} + \frac{1}{2}t^2\ddot{H} + o\,(t^2))\,(\underline{v}_1 + tV_\perp\underline{x} + \frac{1}{2}t^2\underline{y} + o\,(t^2))$$

$$= (\lambda_1 + t\,\dot{\lambda}_1 + \frac{1}{2}t^2\ddot{\lambda}_1 + o\,(t^2))\,(\underline{v}_1 + tV_\perp\underline{x} + \frac{1}{2}t^2\underline{y} + o\,(t^2)) \ . \tag{3.6}$$

Since (3.6) is true for all t, we could expand it and have equalities for its constant, t and t^2 terms individually. Thus for constant term, we have

$$H_0\underline{v}_1 = \lambda_1\underline{v}_1 \ ; \tag{3.7}$$

for t term, we have

$$\dot{H}\underline{v}_1 + H_0 V_\perp\underline{x} = \dot{\lambda}\underline{v}_1 + \lambda_1 V_\perp\underline{x} \tag{3.8}$$

and for t^2 term, we have

$$\dot{H}V_\perp\underline{x} + \frac{1}{2}\ddot{H}\underline{v}_1 + \frac{1}{2}H_0\underline{y} = \dot{\lambda}\,V_\perp\underline{x} + \frac{1}{2}\ddot{\lambda}\underline{v}_1 + \frac{1}{2}\lambda_1\underline{y} \ . \tag{3.9}$$

Now we could express $\dot{\lambda}$ and $\ddot{\lambda}$ in terms of known quantities by performing some simple manipulations. Multiply t term on left by $\underline{v}_1^H$ and yield

$$\underline{v}_1^H(\dot{H}\underline{v}_1 + H_0 V_\perp\underline{x}) = \underline{v}_1^H(\dot{\lambda}\underline{v}_1 + \lambda_1 V_\perp\underline{x}) \tag{3.10}$$

and

$$\underline{v}_1^H\dot{H}\underline{v}_1 + \underline{v}_1^H H_0 V_\perp\underline{x} = \underline{v}_1^H\,\dot{\lambda}\underline{v}_1 + \underline{v}_1^H\,\lambda_1 V_\perp\underline{x} \ . \tag{3.11}$$

Since

$$\underline{v}_1^H H_0 V_\perp \underline{x} = \underline{v}_1^H \lambda_1 V_\perp \underline{x} = 0 \tag{3.12}$$

We then have

$$\dot{\lambda} = \underline{v}_1^H \dot{H} \underline{v}_1 \tag{3.13}$$

Multiply t term on left by $V_\perp^H$ and yield

$$V_\perp^H \dot{H} \underline{v}_1 + \Lambda_\perp \underline{x} = \lambda_1 \underline{x} \tag{3.14}$$

Hence

$$\underline{x} = (\lambda_1 I - \Lambda_\perp)^{-1} V_\perp^H \dot{H} \underline{v}_1 \tag{3.15}$$

Multiply t^2 term on left by $\underline{v}_1^H$ and yield

$$\underline{v}_1^H \dot{H} V_\perp \underline{x} + \frac{1}{2} \underline{v}_1^H \ddot{H} \underline{v}_1 + \frac{1}{2} \underline{v}_1^H H_0 \underline{y} = \dot{\lambda} \underline{v}_1^H V_\perp \underline{x} + \frac{1}{2} \ddot{\lambda} \underline{v}_1^H \underline{v}_1 + \frac{1}{2} \lambda_1 \underline{v}_1^H \underline{y} \tag{3.16}$$

Since for any y

$$\underline{v}_1^H H_0 \underline{y} = \lambda_1 \underline{v}_1^H \underline{y} \quad , \tag{3.17}$$

thus we have

$$\ddot{\lambda} = \underline{v}_1^H \ddot{H} \underline{v}_1 + 2 \underline{v}_1^H \dot{H} V_\perp \underline{x} = \underline{v}_1^H \ddot{H} \underline{v}_1 + 2 \underline{v}_1^H \dot{H} V_\perp (\lambda_1 I - \Lambda_\perp)^{-1} V_\perp \dot{H} \underline{v}_1 \tag{3.18}$$

Note that, as long as that $[\underline{v}_1 \mid V_\perp]$ is unitary and (3.5) holds, $\ddot{\lambda}$ is independent of the choices of $V_\perp$ and $\Lambda_\perp$. By the assumption that λ_1 is simple, (3.13) and (3.18) give the explicit expressions for $\dot{\lambda}$ and $\ddot{\lambda}$ respectively. Since the matrix $(\lambda_1 I - \Lambda_\perp)$ in (3.18) is not invertible when λ_1 is not simple. As mentioned in the beginning of this section, it amounts to the choices of eigenvectors for the case when eigenvalues coalesce, such that, $V_\perp^H \dot{H} \underline{v}_1$ is in the range space of $(\lambda_1 I - \Lambda_\perp)$. Thus a solution of $\underline{x}$ could be

$$\underline{x} = (\lambda_1 I - \Lambda_\perp)^{+} V_\perp^H \dot{H} \underline{v}_1 \tag{3.19}$$

where the superscript '+' denotes pseudo-inverse. Therefore, (3.18) becomes valid after $(\lambda_1 I - \Lambda_\perp)^{-1}$ is replaced by $(\lambda_1 I - \Lambda_\perp)^{+}$. For simplicity of discussion, we assume that

$$\lambda_1 = \lambda_2 = \cdots = \lambda_q \tag{3.20}$$

and $\underline{\phi}_i$, $i=1, \cdots ,q$, are q mutually perpendicular unit norm eigenvectors associated with eigenvalue λ_1.

Proposition 3.1 There exists a choices of mutually perpendicular unit norm eigenvectors $\underline{v}_i$, $i=1, \cdots ,q$, in the space spanned by $\{\underline{\phi}_1\ \underline{\phi}_2\ \cdots\ \underline{\phi}_q\}$ such that for all $i=1, \cdots ,q$, $V_{\perp,i}^H \dot{H} \underline{v}_i$ lies in the range space of $(\lambda_i I - \Lambda_{\perp,i})$ where $V_{\perp,i}$ and $\Lambda_{\perp,i}$ are defined similarly to $V_\perp$ and $\Lambda_\perp$.

Proof. It is easy to show that it suffices to prove that there exists a unitary matrix W , $W \in C^{q \times q}$, such that

$$W^H [\underline{\phi}_1\ \underline{\phi}_2\ \cdots\ \underline{\phi}_q]^H \dot{H} [\underline{\phi}_1\ \underline{\phi}_2\ \cdots\ \underline{\phi}_q] W \tag{3.21}$$

is diagonal. Since matrix $[\underline{\phi}_1\ \underline{\phi}_2\ \cdots\ \underline{\phi}_q]^H \dot{H} [\underline{\phi}_1\ \underline{\phi}_2\ \cdots\ \underline{\phi}_q]$ is hermitian, it is always possible to change (3.21) to diagonal form by performing a unitary transformation. It should be noted that the choice of matrix W is dependent of matrix $\dot{H}$.

[]

4. A second order algorithm.

In this section, we will make use of the properties discussed in the previous section to derive a second order algorithm to solve (2.1). Define

$$H(t) = (e^{Dt} M e^{-Dt})^H (e^{Dt} M e^{-Dt}) \tag{4.1}$$

and let the singular value decomposition of M be

$$M = U \Sigma V^H \tag{4.2}$$

where

$$U = [u_1 \cdots u_n] \tag{4.3}$$

$$V = [v_1 \cdots v_n] \tag{4.4}$$

$$\Sigma = \text{diag}\ \{\sigma_1 \cdots \sigma_n\}\ . \tag{4.5}$$

Then it is easy to get the following equalities

$$H_0 = M^H M \tag{4.6}$$

$$\dot{H} = -DM^H M + 2M^H DM - M^H MD \tag{4.7}$$

$$\ddot{H} = D^2 M^H M - 4DM^H DM + 4M^H D^2 M + 2DM^H MD - 4M^H DMD + M^H MD^2 \tag{4.8}$$

If λ_1 is simple, we have

$$\begin{aligned}\dot{\lambda}_1 &= 2\sigma_1^2(u_1^H D u_1 - v_1^H D v_1) \\ &= 2[d_1 \cdots d_m] \begin{bmatrix} \|P_1 M v_1\|^2 - \|M v_1\|^2 \|P_1 v_1\|^2 \\ \cdot \\ \cdot \\ \cdot \\ \|P_m M v_1\|^2 - \|M v_1\|^2 \|P_m v_1\|^2 \end{bmatrix}\end{aligned}$$

$$= 2\sigma_1^2[d_1 \cdots d_m] \begin{bmatrix} \|P_1 u_1\|^2 - \|P_1 v_1\|^2 \\ \vdots \\ \|P_m u_1\|^2 - \|P_m v_1\|^2 \end{bmatrix} \tag{4.9}$$

and

$$\ddot{\lambda}_1 = [u_1^H D \quad v_1^H D] E \begin{bmatrix} Du_1 \\ Dv_1 \end{bmatrix} \tag{4.10}$$

where

$$P_i = \text{block diag}\,(O_{k_1}, \cdots, O_{k_{i-1}}, I_{k_i}, O_{k_{i+1}}, \cdots, O_{k_m}) , \tag{4.11}$$

$$E = \begin{bmatrix} 4\sigma_1^2 I & -4\sigma_1 M \\ -4\sigma_1 M^H & 2\sigma_1^2 I + 2M^H M \end{bmatrix} +$$

$$2 \begin{bmatrix} 2\sigma_1 U_\perp \Sigma_\perp \\ -V_\perp(\sigma_1^2 I + \Sigma_\perp^2) \end{bmatrix} (\sigma_1^2 I - \Sigma_\perp^2)^{-1} \begin{bmatrix} 2\sigma_1 U_\perp \Sigma_\perp \\ -V_\perp(\sigma_1^2 I + \Sigma_\perp^2) \end{bmatrix}^T , \tag{4.12}$$

$$U_\perp = [u_2 \cdots u_n] , \tag{4.13}$$

$$V_\perp = [v_2 \cdots v_n] , \tag{4.14}$$

and

$$\Sigma_\perp = \text{diag}\,\{\sigma_2, \cdots, \sigma_n\} .$$

Proposition 4.1

$$\lambda_1(t) = \lambda_1 + t < \nabla g\,(0), \underline{d} > + \frac{1}{2} t^2 \underline{d}^T B \underline{d} + o(t^2 \|\underline{d}\|^2) \tag{4.15}$$

where $\underline{d} = [d_1 \cdots d_{m-1}]^T$, $\nabla g\,(0)$ is defined in Section 2 and

$$B = \text{real part of} \begin{bmatrix} P_1 u_1 & \cdots & P_{m-1} u_1 \\ P_1 v_1 & \cdots & P_{m-1} v_1 \end{bmatrix}^H E \begin{bmatrix} P_1 u_1 & \cdots & P_{m-1} u_1 \\ P_1 v_1 & \cdots & P_{m-1} v_1 \end{bmatrix} . \tag{4.16}$$

Furthermore, B is non-negative definite.

[]

The first three terms in the right hand side of (4.15) gives a second order model of $\lambda_1(t)$ when λ_1 is simple and $t\|d\|$ is small. Since B is non-negative definite, the solution with minimum norm for the model is

$$t \| d \| = -B^{+} \nabla g(0) \tag{4.17}$$

and also $-B^{+}\nabla g(0)$ is a descent direction of $g(\underline{d})$ at $\underline{d}=0$, where B^{+} denotes the pseudo-inverse of B. We now state the algorithm to solve (2.1).

Algorithm 4.1

Step 1.

Data $M_0 = M$, $D_0 = 0$ ($\underline{d}_0 = 0$).

$k = 0$.

Step 2.

Set $M_{k+1} = e^{D_k} M_k e^{-D_k}$.

If the largest singular value of M_{k+1} is simple, define search direction $\underline{h}$ to be $-B^{+}\nabla g(0)$, otherwise define search direction $\underline{h}$ to be $-Nr(\mathrm{co}\nabla_2)$ where ∇_2, B and $\nabla g(0)$ are defined in terms of M_{k+1}.

Step 3.

Perform line search to find the step size α.

Step 4.

$\underline{d}_{k+1} = \underline{d}_k + \alpha \underline{h}$ (D_{k+1} is therefore updated).

Set $k = k+1$, go to step 2.

[]

Appendix A. Counterexample of $\mu(M) \neq \inf_{D \in d} (e^D M e^{-D})$

Let $a = (1-(\frac{1}{3})^{1/2})^{1/2}$, $b = \frac{1}{2^{1/2}}$ and

$$M_1 = \begin{bmatrix} a & 0 \\ ab & ab \\ ab & abi \\ (1-2a^2)^{1/2} & -\frac{a^2(1+i)}{2(1-2a^2)^{1/2}} \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 0 & a \\ ab & -ab \\ ab & -abi \\ \frac{a^2(1-i)}{2(1-2a^2)^{1/2}} & (1-2a^2)^{1/2} \end{bmatrix}.$$

Define $M = M_1 M_2^H$ and structure $k = (1,1,1,1)$, then $\bar{\sigma}(M) = 1$ and

$$\bar{H}_1 = a^2 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \bar{H}_2 = a^2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \bar{H}_3 = a^2 \begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix}.$$

It is easy to check that ∇_2 is a circle with radius a^2 centered at origin. Thus $\nabla_2 \neq \mathrm{co}\nabla_2$, $0 \in \mathrm{co}\nabla_2$ and $0 \notin \nabla_2$. Therefore $\mu(M) < 1$ but

$$\inf_{D \in d} (e^D M e^{-D}) = 1$$

For this example, by using the formula in, [5] we can show that $\mu(M) > 0.87$.

References

[1] J.C. Doyle, "Analysis of Feedback Systems with Structured Uncertainties," *Proceedings of IEE* vol. 129, no. 6, pp. 242-250 (November 1982).

[2] J.C. Doyle, J.E. Wall, and G. Stein, "Performance and Robustness Analysis for Structured Uncertainty," *Proc. 21st IEEE Conf. on Decision and Control,* pp. 629-636 (December 1982).

[3] J.C. Doyle, *Private communication.* 1984.

[4] M.G. Safonov and J.C. Doyle, "Minimizing Conservativeness of Robustness Singular Values," pp. 197-207 in *Multivariable Control,* ed. S.G. Tzafestas, D. Reidel Publishing Company (1984).

[5] M.K.H. Fan and A.L. Tits, "Characterization and Efficient Computation of the Structured Singular Value," *IEEE Trans. on Automatic Control* vol. AC-31, no. 8, (1986).