

**User's Guide For MUSOL2: A  
Package For Computing The  
Structured Singular Value Or Its  
Upper Bound**

**by**

**M. K.H. Fan**



# USER'S GUIDE FOR MUSOL2: A PACKAGE FOR COMPUTING THE STRUCTURED SINGULAR VALUE OR ITS UPPER BOUND \*

Michael K.H. Fan  
Systems Research Center  
University of Maryland, College Park, MD 20742

## Abstract

The concept of structured singular value was introduced by Doyle [1] as a tool for the 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 very nicely complements the  $H^\infty$  approach to control system design. This report is the user's guide for Version 1.0 of MUSOL2, a set of Fortran and C subroutines designed to compute a good upper bound for (in some cases the exact value of) the structured singular value of a square complex constant matrix with respect to a structure accepting for both real scalar and complex block uncertainties. The driving routine of this package is a Fortran subroutine. Details on the theoretical foundations for the algorithm employed can be found in [2].

## 1 Purpose

The concept of structured singular value was introduced by Doyle [1] as a tool for the 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 very nicely complements the  $H^\infty$  approach to control system design. MUSOL2 is a set of Fortran and C subroutines designed to compute a good upper bound for (in some cases the exact value of) the structured singular value of a square complex constant matrix with respect to a structure accepting for both real scalar and complex block uncertainties. The driving routine of this package is a Fortran subroutine. Details on the theoretical foundations for the algorithm employed can be found in [2].

Given an  $n \times n$  complex matrix  $A$  and two nonnegative integers  $m_r$  and  $m_C$ , with  $m := m_r + m_C \leq n$ , a *block structure*  $\mathcal{K}$  of dimensions  $(m_r, m_C)$  associated with  $A$  is an  $m$ -tuple

---

\*This research was supported by the National Science Foundation under grants No. DMC-84-51515 and CDR-85-00108.

of positive integers

$$\mathcal{K} := (k_1, \dots, k_{m_r} ; k_{m_r+1}, \dots, k_{m_r+m_c})$$

such that  $\sum_{i=1}^m k_i = n$ . Given a block structure  $\mathcal{K}$ , consider the family of block diagonal  $n \times n$  matrices

$$\mathcal{X}_{\mathcal{K}} := \{\text{block diag } (\delta_1 I_{k_1}, \dots, \delta_{m_r} I_{k_{m_r}}, \Delta_1, \dots, \Delta_{m_c}) : \delta_i \in \mathbb{R}, \Delta_i \in \mathbb{C}^{k_{m_r+i} \times k_{m_r+i}}\}$$

where, here and in the sequel, for any integer  $k$ ,  $O_k$  denotes  $k \times k$  zero matrix and  $I_k$   $k \times k$  identity matrix. The *structured singular value*  $\mu_{\mathcal{K}}(A)$  of a complex  $n \times n$  matrix  $A$  with respect to block structure  $\mathcal{K}$  is

$$\mu_{\mathcal{K}}(A) := \begin{cases} \left( \min_{\Delta \in \mathcal{X}_{\mathcal{K}}} \{\bar{\sigma}(\Delta) : \det(I - \Delta A) = 0\} \right)^{-1} & \text{if } \det(I - \Delta A) = 0 \text{ for some } \Delta \in \mathcal{X}_{\mathcal{K}} \\ 0 & \text{otherwise.} \end{cases}$$

Assume first that

$$k_i = 1, \quad \forall i \leq m_r. \quad (1)$$

Consider the projection matrices  $P_i$ ,  $i = 1, \dots, m$ , defined by

$$P_i := \text{block diag } (O_{k_1}, \dots, O_{k_{i-1}}, I_{k_i}, O_{k_{i+1}}, \dots, O_{k_m}).$$

For  $i = 1, \dots, m$ , let

$$Q_i := A^H P_i A$$

and, for  $i = 1, \dots, m_r$ , let

$$R_i := j(P_i A - A^H P_i)$$

where  $j = \sqrt{-1}$  and the superscript “ $H$ ” indicates complex conjugate transpose.

Suppose that  $\alpha \in \mathbb{R}^m$ ,  $\beta \in \mathbb{R}^{m_r}$  and  $\gamma \in \mathbb{R}$ . Define

$$H(\alpha, \beta, \gamma) := \sum_{i=1}^m \alpha_i (Q_i - \gamma R_i) + \sum_{i=1}^{m_r} \beta_i B_i.$$

Also, let  $\eta(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^{m_r} \rightarrow \mathbb{R}^+$  be defined by

$$\eta(\alpha, \beta) := \begin{cases} \max_{\gamma \geq 0} \left\{ \sqrt{\gamma} : \bar{\lambda}(H(\alpha, \beta, \gamma)) \geq 0 \right\} & \text{if } \bar{\lambda}(H(\alpha, \beta, \gamma)) \geq 0 \text{ for some } \gamma \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where  $\bar{\lambda}$  is the largest eigenvalue of its matrix-valued argument.

For any given  $\alpha_{max} \geq 1$ ,  $\beta_{max} \geq 0$ , MUSOL2 computes the quantity  $\hat{\mu}_{\mathcal{K}}(A)$  defined by

$$\hat{\mu}_{\mathcal{K}}(A) := \min_{\alpha, \beta} \{ \eta(\alpha, \beta) : 1 \leq \alpha_i \leq \alpha_{max}, \forall i \leq m, |\beta_i| \leq \beta_{max}, \forall i \leq m_r \} \quad (2)$$

It is shown in [2] that  $\hat{\mu}_{\mathcal{K}}(A)$  is an upper bound of the structured singular value of  $A$  with respect to block-structure  $\mathcal{K}$ , i.e.,  $\mu_{\mathcal{K}}(A) \leq \hat{\mu}_{\mathcal{K}}(A)$ , and that the equality holds if  $\alpha_{max} = \beta_{max} = \infty$  and either  $m_r = 0, m_C \leq 3$  or  $m_r = 1, k_1 = 1, m_C \leq 1$ . Furthermore, numerical experiments show that the equality often holds if  $\alpha_{max} = \beta_{max} = \infty, m_r = 0$  and  $m_C > 3$ .

For the case that (1) does not hold, MUSOL2 can still be used to obtain an upper bound of the structured singular value. This is done by considering the corresponding *augmented* structure. For example, if the given block-structure is (2,3 ; 2,1,4), one then can use the structure (1,1,1,1,1 ; 2,1,4) and apply MUSOL2.

## 2 Description of Algorithm

In this section we give an outline of the algorithm used to compute  $\hat{\mu}_{\mathcal{K}}(A)$  defined in (2). For any  $\gamma \geq 0$ , define

$$\psi(\gamma) := \min_{\alpha, \beta} \left\{ \bar{\lambda}(H(\alpha, \beta, \gamma)) : 1 \leq \alpha_i \leq \alpha_{max}, \forall i \leq m, |\beta_i| \leq \beta_{max}, \forall i \leq m_r \right\}. \quad (3)$$

It is shown in [2] that  $\psi(\cdot)$  is a strictly decreasing function. Furthermore,  $\hat{\mu}_{\mathcal{K}}(A) = 0$  if  $\psi(0) \leq 0$  while if this is not the case  $\psi(\gamma) = 0$  has a *unique* solution at  $\gamma = \hat{\mu}_{\mathcal{K}}^2(A)$ .

Algorithm 1 below solves  $\psi(\gamma) = 0$  by using an enhanced bisection method based on a certain function  $\pi$ . This function (defined in Algorithm 2) has the same zero as  $\psi$  but has the additional property that,  $\hat{\mu}_{\mathcal{K}}(A) \leq \gamma + \pi(\gamma)$  whenever  $\hat{\mu}_{\mathcal{K}}(A) \leq \gamma$ . This feature makes the bisection method converge to the solution faster. Algorithm 2 is based on the cutting plane method [3] with the modification that, in each iteration, the set  $S$  (defined in Algorithm 2) contains only those  $x$ 's which are active in the linear programming problem (see below).

**Algorithm 1.** (Computation of  $\hat{\mu}_{\mathcal{K}}(A)$ )

- Step 1.* Given  $\epsilon > 0$ .  
Set  $\gamma_l = 0$  and  $\gamma_u = \bar{\sigma}^2(A)$ .
- Step 2.* If  $\pi(\gamma_l) \leq 0$ , set  $\gamma_u = 0$  and go to Step 5.  
Otherwise, set  $\gamma_u = \gamma_u + \pi(\gamma_u)$ .
- Step 3.* If  $(\sqrt{\gamma_u} - \sqrt{\gamma_l}) \leq \epsilon \cdot \max(1, \sqrt{\gamma_u})$ , go to Step 5.
- Step 4.* Set  $\gamma = (\gamma_l + \gamma_u)/2$ .  
If  $\pi(\gamma) \geq 0$ , set  $\gamma_l = \gamma$  and go to Step 3.  
Otherwise, set  $\gamma_u = \gamma + \pi(\gamma)$  and go to Step 3.

*Step 5.* Stop.

We have

$$\sqrt{\gamma_u} - \epsilon \cdot \max(1, \sqrt{\gamma_u}) \leq \hat{\mu}_K(A) \leq \sqrt{\gamma_u}. \quad (4)$$

**Algorithm 2.** (Computation of  $\pi(\gamma)$ )

*Step 1.* Given  $\alpha_{max} \geq 1$  and  $\beta_{max} \geq 0$ .

Set  $S$  to be the empty set and  $k = 0$ .

Set  $\alpha^k \in \mathbb{R}^m$  and  $\beta^k \in \mathbb{R}^{m_r}$  arbitrarily with components satisfying the bounds in (2).

*Step 2.* Set  $\psi_u = \bar{\lambda}(H(\alpha^k, \beta^k, \gamma))$ .

If  $\psi_u \leq 0$ , set

$$\pi(\gamma) = \frac{\psi_u}{\max_{1 \leq i \leq m} \alpha_i^k}$$

and return.

Otherwise, let  $x$  be any unit norm eigenvector corresponding to  $\psi_u$ .

Set  $S = S \cup \{x\}$ .

*Step 3.* Let  $(\alpha^{k+1}, \beta^{k+1})$  solve the following linear programming (LP) problem:

$$\psi_l := \min_{\alpha, \beta} \max_{x \in S} \left\{ \left\langle \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \begin{bmatrix} u \\ v \end{bmatrix} \right\rangle : 1 \leq \alpha_i \leq \alpha_{max}, \forall i \leq m, \right. \\ \left. |\beta_i| \leq \beta_{max}, \forall i \leq m_r \right\}$$

where

$$u := \begin{bmatrix} x^H(Q_1 - \gamma P_1)x \\ \vdots \\ x^H(Q_m - \gamma P_m)x \end{bmatrix}, \quad v := \begin{bmatrix} x^H R_1 x \\ \vdots \\ x^H R_{m_r} x \end{bmatrix}.$$

*Step 4.* If  $\psi_l \geq 0$ , set  $\pi(\gamma) = \psi_l$  and return.

Otherwise, let  $S$  contain only those  $x$ 's which are active in the LP problem of Step 3.

Set  $k = k + 1$  and go to Step 2.

### 3 Calling Sequence

SUBROUTINE MUSOL2( AR, AI, LDA, NA, MR, MC, K, ALPHAM, BETAM,  
ALPHA, BETA, EPS, XIU, PRT, INFORM)

INTEGER LDA, NA, MR, MC, PRT, INFORM  
 INTEGER K(MR+MC)  
 DOUBLE PRECISION ALPHAM, BETAM, EPS, XIU  
 DOUBLE PRECISION AR(LDA,NA), AI(LDA,NA), ALPHA(MR+MC), BETA(MR)

#### 4 Input Parameters

- AR** is a double precision array of declared dimension (LDA,NA) that contains the real part of  $A$ .
- AI** is a double precision array of declared dimension (LDA,NA) that contains the imaginary part of  $A$ .
- LDA** is the declared row dimension of  $A$  (LDA must be at least 1).
- NA** is the parameter  $n$ , i.e., the size of  $A$ . NA must be positive.
- MR** is the parameter  $m_r$ , i.e., the number of real blocks in the structure  $\mathcal{K}$ . MR must be nonnegative.
- MC** is the parameter  $m_c$ , i.e., the number of complex blocks in the structure  $\mathcal{K}$ . MC must be nonnegative. Furthermore, MR+MC must be positive and less than or equal to NA.
- K** is an integer array of dimension MR+MC that indicates the block structure  $\mathcal{K}$ . All the components of K must be positive. Furthermore,  $K(I)$  must be 1 for  $I \leq \text{LE}$ . MR and  $K(1) + \dots + K(\text{MR} + \text{MC})$  must be equal to NA.
- ALPHAM** is the parameter  $\alpha_{max}$ . ALPHAM must be no less than 1.
- BETAM** is the parameter  $\beta_{max}$ . BETAM must be nonnegative.
- EPS** is the parameter  $\epsilon$ , which determines the accuracy of the result. EPS must be positive and it should be chosen as the largest possible acceptable value.  $\text{EPS} = 10^{-2}$  seems to be a good tradeoff between accuracy and speed of the computation.
- PRT** indicates the amount of intermediate output desired. The printout is described in Section 7. All output is written to the standard output. For each value of PRT, the output consists of that indicated below for the given value and for the lower values of PRT (except 0).

**PRT**      **Definition**

- 0 No output.
- 1 The final solution only.
- 2 The values of  $\sqrt{\gamma_l}$  and  $\sqrt{\gamma_u}$  before Step 3 of Algorithm 1 is performed.
- 3 The values of  $\psi_l$  and  $\psi_u$  before Step 4 of Algorithm 2 is performed.

## 5 Output Parameters

**INFORM** indicates the result of MUSOL2. The possible values of **INFORM** are:

- 0 Normal return. MUSOL2 computes the value of  $\hat{\mu}_{\mathcal{K}}(A)$  successfully.
- 1 An input parameter is invalid.

**XIU** if **INFORM**=0, contains the final value of  $\sqrt{\gamma_u}$  for which (4) holds.

**ALPHA** is a double precision array of dimension **MR**+**MC**. if **INFORM**=0, contains the best estimate of the solution of  $\alpha$ .

**BETA** is a double precision array of dimension **MR**. if **INFORM**=0, contains the best estimate of the solution of  $\beta$ .

## 6 Auxiliary Subroutines

The subroutines associated specifically with the MUSOL2 package are as follows:

**MUU**          **PI**          **LP**          **SMALLS**

MUSOL2 also uses routines

**FM01AD**      **LA01BD**      **LA01CD**      **LA01ED**

from Harwell Subroutine Library [4] for solving LP problems. Thus a potential user needs to have access to these routines. The routine **LP** is largely inspired from the **RATTLE** routine **Mlinproc** in the **DELIGHT** system [5].

## 7 Description of Printed Output

The intermediate printout produced by MUSOL2 is defined as follows:

**XIL**      is  $\sqrt{\gamma_l}$ .

**XIU**      is  $\sqrt{\gamma_u}$ .

PSIL is  $\psi_l$ .

PSIU is  $\psi_u$ .

ALPHA is  $\alpha$ .

BETA is  $\beta$ .

## 8 Example Program and Output

This section contains a source listing and the computed results for a sample main program that calls MUSOL2 to compute the upper bound of the structured singular value of the matrix  $A$  given by

$$A = \begin{bmatrix} 0.3 & 1.3 & -0.7 & 0.4 & 0.2 \\ 1.7 & 2.1 & 1.9 & 0.5 & -2.6 \\ -0.4 & 1.7 & 1.6 & -0.1 & 0.0 \\ 0.6 & 2.0 & 0.3 & 2.4 & 0.0 \\ 1.4 & -0.2 & 1.2 & 1.8 & 0.6 \end{bmatrix} + j \begin{bmatrix} 1.8 & 0.0 & 0.5 & 0.4 & -0.3 \\ 1.0 & 1.0 & 0.3 & -3.6 & -3.6 \\ -0.4 & 3.3 & 0.4 & 2.2 & 0.8 \\ -0.9 & -1.6 & -0.9 & -2.4 & -1.0 \\ -1.3 & 1.1 & -3.0 & 0.5 & -0.8 \end{bmatrix}$$

where  $j$  denotes  $\sqrt{-1}$ . Input parameters are  $m_r = 2$ ,  $m_C = 2$ ,  $\mathcal{K} = (1, 1; 2, 1)$ ,  $\alpha_{max} = \beta_{max} = 10^6$  and  $\epsilon = 10^{-3}$ . Furthermore, the parameter PRT is set to 2 for requesting some intermediate output. The computed solution is

$$\sqrt{\gamma_u} = 7.67922, \quad \alpha = \begin{bmatrix} 1.0 \\ 66839.2 \\ 62052.7 \\ 95357.4 \end{bmatrix}, \quad \beta = \begin{bmatrix} 100000.0 \\ 90796.3 \end{bmatrix}.$$

### 8.1 Example Program

```
C
C=====
C
C EXAMPLE PROGRAM FOR SUBROUTINE MUSOL2.
C VERSION 1.0, AUGUST 1988.
C
C=====
C
C      INTEGER          LDA, NA, MR, MC, PRT, INFORM
C      INTEGER          K(4)
C      DOUBLE PRECISION ALPHAM, BETAM, EPS, XIU
C      DOUBLE PRECISION AR(5,5), AI(5,5), ALPHA(4), BETA(2)
C
C      LDA = 5
C      NA = 5
```

```

MR = 2
MC = 2
PRT = 2
C
K(1) = 1
K(2) = 1
K(3) = 2
K(4) = 1
C
ALPHAM = 1.0D6
BETAM = 1.0D6
C
EPS = 1.0D-3
C
C READ MATRICES AR AND AI
C
READ (5,*) ((AR(I,J),J=1,NA), I=1,NA)
READ (5,*) ((AI(I,J),J=1,NA), I=1,NA)
C
C WRITE MATRICES AR AND AI
C
WRITE (6,100) ((AR(I,J),J=1,NA), I=1,NA)
WRITE (6,101) ((AI(I,J),J=1,NA), I=1,NA)
WRITE (6,102)
C
C CALL MUSOL2
C
CALL MUSOL2( AR, AI, LDA, NA, MR, MC, K, ALPHAM, BETAM,
*           ALPHA, BETA, EPS, XIU, PRT, INFORM )
C
100  FORMAT(/ 5H AR =/ (1X, 5E10.2))
101  FORMAT(/ 5H AI =/ (1X, 5E10.2))
101  FORMAT(/)
C
C END OF THE EXAMPLE PROGRAM OF MUSOL2
END

```

## 8.2 Output

```

AR =
  0.30e+00  0.13e+01 -0.70e+00  0.40e+00  0.20e+00
  0.17e+01  0.21e+01  0.19e+01  0.50e+00 -0.26e+01
 -0.40e+00  0.17e+01  0.16e+01 -0.10e+00  0.00e+00
  0.60e+00  0.20e+01  0.30e+00  0.24e+01  0.00e+00
  0.14e+01 -0.20e+00  0.12e+01  0.18e+01  0.60e+00

```

```

AI =
  0.18e+01  0.00e+00  0.50e+00  0.40e+00 -0.30e+00
  0.10e+01  0.10e+01  0.30e+00 -0.36e+01 -0.36e+01
 -0.40e+00  0.33e+01  0.40e+00  0.22e+01  0.80e+00
 -0.90e+00 -0.16e+01 -0.90e+00 -0.24e+01 -0.10e+01

```

-0.13e+01 0.11e+01 -0.30e+01 0.50e+00 -0.80e+00

XIL= 0.00000e+00	XIU= 7.92331e+00
XIL= 5.60263e+00	XIU= 7.92331e+00
XIL= 6.86179e+00	XIU= 7.92331e+00
XIL= 7.41158e+00	XIU= 7.92331e+00
XIL= 7.67172e+00	XIU= 7.92331e+00
XIL= 7.67172e+00	XIU= 7.73714e+00
XIL= 7.67172e+00	XIU= 7.70344e+00
XIL= 7.67172e+00	XIU= 7.68697e+00
XIL= 7.67172e+00	XIU= 7.67922e+00

SOLUTION: XIU = 7.67922e+00  
ALPHA( 1) = 1.00000e+00  
ALPHA( 2) = 6.68392e+04  
ALPHA( 3) = 6.20527e+04  
ALPHA( 4) = 9.53574e+04  
BETA( 1) = 1.00000e+06  
BETA( 2) = 9.07963e+04

## References

- [1] J.C. Doyle, "Analysis of Feedback Systems with Structured Uncertainties," *Proc. IEE-D* 129 (1982), 242-250.
- [2] M.K.H. Fan, "On Computing the Structured Singular Value of a Square Transfer Matrix," (*in preparation*) (1988).
- [3] J.E. Kelley, "The Cutting-Plane Method for Solving Convex Programs," *SIAM Journal* 8 (December 1960), 703-712.
- [4] "Harwell Subroutine Library," Library Reference Manual, Harwell, England, 1982.
- [5] W.T. Nye, E. Polak, A. Sangiovanni-Vincentelli & A.L. Tits, "DELIGHT: An Optimization-Based Computer-Aided Design System," *Proceedings of the 1981 IEEE International Symposium on Circuits and Systems* (April 1981).