Two Challenging Problems in Control Theory

Minyue Fu
School of Electrical Engineering and Computer Science
University of Newcastle, NSW 2308 Australia
Email: minyue.fu@newcastle.edu.au

Abstract

This chapter discusses two important yet challenging theoretical problems for control systems, namely, the static output feedback stabilization problem and the structured singular value problem. Both problems are well known in the control literature and have been intensively studied. This article introduces several known results related to the computational complexities of these problems, with the aim to encourage new research on them.

Keywords: Static output feedback control, robustness analysis, robust control, computational complexity, linear systems.

1 Introduction

Since the classical work of Bode, control theory has been around for over 60 years, making major contributions to a wide range of engineering, scientific, medical and social applications. To some extend, the field has reached a certain degree of maturity and many new branches of control theory have emerged which penetrate deeply into neighboring disciplines. Yet some of the classical control problems remain unsolved. The purpose of this chapter is to discuss two of such problems.

The two classical control problems we study here are the static output feedback stabilization problem and the structured singular value problem, both being well known in the literature with a vast amount of work devoted to them. This chapter introduces some of the known results on the computational complexities of these problems. Our aim is to raise new interest in these problems so that new and better algorithms or solutions can be found for solving them.

Due to the fact that these two problems are quite different in nature, we will discuss them separately. But their common thread is their algorithmic difficulties. For this reason, we will start with some brief discussion on the theory of computational complexity.

2 Basics of Computational Complexity

In theory of computational complexity, numerical problems are classified according to the computational time required for solving them on a Turing machine which is a sequential digital computer with Boolean operations.

Only decision problems are considered. For example, instead of asking "what is the minimum value of a given quadratic function $y = f(x)$", we ask whether the value of quadratic function can be less than a given value. The former is an optimization problem whereas the latter is a decision problem. The reason is that the two questions have very similar levels of computational complexity. More precisely, if there is an efficient algorithm for the decision problem, the optimization problem can be easily solved to an arbitrarily high precision using a bisection method. This method starts with some pre-assigned lower bound $\underline{y}$ and upper bound $\bar{y}$ and repeat the following steps until the two bounds are sufficiently close:

- Take $y^* = (\underline{y} + \bar{y})/2$;
- Check if there exists x such that $f(x) \leq y^*$. If yes, reset $\bar{y} = y^*$, else reset $\underline{y} = y^*$.

Note that the bisection algorithm solves the decision problem repeatedly, but two bounds converge very quickly. A lot of algorithms more efficient than the bisection algorithm can also be used, depending on the given problem.

The complexity class P denotes a class of decision problems which can be solved by a deterministic Turing machine in polynomial time. The class NP denotes a class of decision problems which can be solved by a nondeterministic Turing machine in polynomial time, including P as a subclass. The exact definitions of these two classes are involved and can be found in Garey and Johnson [15] and Papadimitroius and Steiglitz [23]. Roughly speaking, every P problem has a deterministic polynomial time algorithm, and every NP problem has deterministic exponential time algorithm. An algorithm is called deterministic if it gives a definitive answer (yes or no) for the given problem data. Most commonly used algorithms are of this kind.

The term polynomial time algorithm (or polynomial algorithm for short) means an algorithm which requires only a polynomial number of steps and polynomial storage to excute on a Turing machine. The data (called instance) of a decision problem are assumed to be rational to avoid the complexity issues for real numbers.

Checking whether a given matrix is nonsingular is a simple example of P problems. The formal statement of the problem is as follows:

Instance: Given an $n \times n$ matrix A with rational entries.

Question: Is $\det(A) \neq 0$?

Examples of P problems are abundant in control theory: controllability and observability of a linear system (the decision problems are whether the system is controllable and whether the system is observable), LQG control and H_∞ control (the decision problems are whether there exists a controller so that the LQG cost or H_∞ norm of the closed-loop system is less than a given level). In fact, when formulated as a decision problem, most classical control algorithms are in P.

A simple example of NP problem, which is not known to be P, is the maximization of a convex quadratic function. A convex quadratic function has the form $f(x) = x^T Qx + b^T x + c$, where $Q \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}$ are given data and $x \in \mathbf{X}^n$ with

$$\mathbf{X}^n = \{x \in \mathbb{R}^n : |x_i| \leq 1\}$$

which is also given. Minimizing such a quadratic function is known to be a P problem and can be easily solved by using algorithms such as Newton gradient method.

However, maximizing such a function is much harder. To check the computational complexity of the problem, we need to formulate it as a decision problem, as follows:

Instance: Given $Q \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$ and α with rational entries.

Question: Does there exist $x \in \mathbf{X}^n$ such that $f(x) = x^T Qx + b^T x + c > \alpha$?

The parameter α is introduced so that the maximization problem becomes a decision problem. As explained earlier, a bisection method can be applied to find the maximum with any required level of accuracy.

To see that the decision problem above is in NP, we simply note that $\mathbf{X}^n$ is a n-dimensional box and the maximum of a convex function occurs at one of its vertices. Since there are 2^n vertices and checking whether $f(x) > \alpha$ at a given vertex is a simple problem (P problem), the decision problem can be solved with exponential time complexity, hence a NP problem.

Examples of other classical NP problems which are not known to be P include the traveling salesman problem, the maximum cut problem, and the 3-SAT problem [15, 23].

It is generally believed that NP$\neq$ P, although it has been a great challenge in combinatoric optimization for the last several decades to prove or disapprove it. A decision problem is called NP-complete if it lies in NP and every NP problem can be transformed in polynomial time into this problem. All the non-P examples of NP above are NP-complete. All NP-complete problems are equivalent in the sense that they can be reduced into each other in polynomial time.

A problem is called NP-hard if an NP-complete problem can be reduced to this problem in polynomial time. An NP-hard problem does not have to be in NP. In control theory, the structured singular value problem, which is to be studied in this chapter, is known to be NP-hard [3]. So, an NP-hard is at least as "hard" as a NP-complete problem.

In proving that a control problem (or problems in other disciplines) is NP-hard, we usually make no assumption for the problem data to be rational. Instead, we typically analyze the special case where the problem data are rational and show that it is NP-complete. As a result of it, the general case where the data are allowed to be real is NP-hard.

3 Static Output Feedback Stabilization Problem

Consider a linear time-invariant system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{aligned} \tag{3.1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, $y(t) \in \mathbb{R}^r$ is the measured output, and A, B and C are known matrices with appropriate dimensions. The problem of static output feedback stabilization is to find a constant matrix $K \in \mathbb{R}^{m\times r}$ such that the controller

$$u(t) = Ky(t) \tag{3.2}$$

stabilizes (3.1), namely, the closed-loop matrix $A + BKC$ is Hurwitz, i.e., its eigenvalues are all in the open left half plane.

From algorithmic point of view, we want to find an algorithm which can determine, in a "computationally tractable" way, whether any given system (3.1) admits a static output feedback stabilizer (3.2). By "computationally tractable", we mean "with polynomial complexity."

This seemingly simple problem has been challenging the control researchers ever since the time of Bode. It is well known that dynamic output feedback stabilization is a trivial problem to solve. The necessary and sufficient condition for stabilizability is that [27] the system the pair (A, B) is stabilizable and the pair (A, C) is detectable. However, the restriction of static output feedback makes the problem far more difficult. Despite of many attempts to solve this problem, no satisfactory solutions have been found. Early work mainly focused on finding algorithms which can lead to stabilization. However, such algorithms typically require strong conditions on the system or fail to work in general. In recent years, focus has been shifted to considering the potential inherent computational difficulties of the problem. Three such attempts are to be discussed in this chapter.

At this point, the reader may wonder why we are so obsessed with the static output feedback stabilization problem. The main reason is that many important control problems are inherently related to this problem. Anyone familiar with the modern control theory can point to many wonderful algorithms for control design, such as LQG design and H_∞ control. But often the time, these algorithms require the controller to be of full order. In many applications, low-order controllers are preferred. So there is a serious gap between good known control algorithms and practical control requirements. If we restrict the order of the controller to be a low-order one, the problem becomes hard, just as the static output feedback stabilization case. Here we list two such problems as examples.

The first one is the fixed-order control problem. In this case, we consider the following system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_1 w(t) + B_2 u(t) \\ z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t) \\ y(t) &= C_2 x(t) + D_{21} w(t) \end{aligned} \tag{3.3}$$

where $x(t), u(t)$ and $y(t)$ are as before, $w(t) \in \mathbb{R}^p$ and $z(t) \in \mathbb{R}^q$ are exogenous input and controlled output, separately, and $A, B_1, B_2, C_1, C_2, D_{11}, D_{12}$ and D_{21} are known matrices of appropriate dimensions. The controller takes the following form:

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c y(t) \\ u(t) &= C_c x_c(t) + D_c y(t) \end{aligned} \tag{3.4}$$

where $x_c(t) \in \mathbb{R}^{n_c}$ is the state of the controller with its order n_c given, and A_c, B_c, C_c and D_c are controller parameters to be designed. The control objective is to find a controller (3.4) such that the H_∞ norm of the closed-loop system, i.e., the transfer function from $w(t)$ to $z(t)$, is less than a given level, say, one. Accordingly, we seek a computationally tractable algorithm to determine if such a controller exists for any given system (3.3) and n_c.

It is obvious that when $n_c = n$, we have the so-called full-order control problem, which has been well studied and there exist simple algorithms of polynomial complexity; see, e.g., [7]. Although we do not go into details here, but it can be shown that the problem is still relatively easy to deal with when $n_c = n - 1$. The problem becomes difficult when $n_c < n - 1$. The static output feedback case where $n_c = 0$ is a special but important case.

We remark here that the level of difficulties of the above problem does not depend on the H_∞ performance requirement. The same difficulty remains when another performance index, for example, quadratic cost as used in classical LQG control, is used, or even when stabilization is the mere objective.

The second example related to static output feedback stabilization is static output feedback D-stabilization. For the same system (3.1) and controller (3.2), the objective here is to find K such that the closed-loop system has eigenvalues distributed in the desired region D. By distribution, we mean that the region D can consist of several disjoint sub-regions and a fixed number of eigenvalues are required in each sub-region. Obviously, if D is the open-left half plane involving no sub-region, the D-stabilization problem is the usual stabilization problem. On the other hand, if D consists of n fixed points in the complex plane, the D-stabilization becomes a pole-placement problem. The purpose of D-stabilization is to provide some guarantee on the closed-loop performance.

From computational complexity point of view, the level of difficulty of the D-stabilization problem does not appear to depend on D. That is, the problem appears equally hard for most regions. Also, the problem does not appear to become easier if a fixed-order controller (3.4) is used when $n_c < n - 1$.

We now set out to discuss three attempts for the static output feedback stabilization problem.

3.1 Static Output Feedback Stabilization with Confined Feedback Gain

This result is due to Blondel and Tsitsiklis [1]. Unlike the standard output feedback stabilization problem, they restrict the feedback gain K to be in a confined region. That is, given $\underline{k}_{ij}$ and $\bar{k}_{ij}$, $1 \le i \le m$, $1 \le j \le r$, $K = \{k_{ij}\}$ is required to have

$$\underline{k}_{ij} \le k_{ij} \le \bar{k}_{ij} \tag{3.5}$$

The result is as follows:

Theorem 3.1 *[1] The following problem in NP-hard:*

Instance: *Positive integers n, m and r, $n \times n$ matrix A, $n \times m$ matrix B and $r \times n$ matrix C with rational coefficients, and rational numbers $\underline{k}_{ij}$ and $\bar{k}_{ij}$, $1 \le i \le m$, $1 \le j \le r$.*

Question: *Does there exist a real matrix $K = \{k_{ij}\}$ satisfying $\underline{k}_{ij} \le k_{ij} \le \bar{k}_{ij}$ and that $A + BKC$ is Hurwtiz?*

Remark: At first glance, the result above appears to have shown that the static output feedback stabilization problem is NP-hard and our quest for computational complexity of the problem is over. However, this is far from true. The reality is that the restriction on the control gain (3.5) has made the problem far harder than necessary. Indeed, it is shown in [1] that the state feedback stabilization problem (i.e., when $C = I$) is still NP-hard when the same restriction (3.5) is imposed. Not only this, a seemingly much simpler problem, the so-called stable matrix in interval family problem, which has $A = 0, B = I$ and $C = I$, is also shown to be NP-hard. Actually, the way that Theorem 3.1 is proved is to show that the special case of stable matrix in interval family problem is NP-hard. As we all know, state feedback stabilization with restriction on the feedback gain is a simple classical problem. The reason the problem becomes very hard when restriction (3.5) is imposed is that the algorithm we look for needs to be able to check for any given box of K, where the original stabilization problem only needs to find one stabilizing K.

3.2 Matrix inequality Approach

Denote by $\mathbf{R}_s^n$ the set of $n \times n$ symmetric and real matrices. The problem we consider here is stated as follows: Given two affine mappings $L_1(\cdot), L_2(\cdot) : \mathbf{R}_s^n \to \mathbf{R}_s^n$, find positive definite matrices $X, Y \in \mathbf{R}_s^n$ such that

$$L_1(X) < 0, \quad L_2(Y) < 0, \quad XY = I \tag{3.6}$$

The two inequalities above are linear matrix inequalities but the equality above imposes a bilinear constraints on X and Y. This is a special case of bilinear matrix inequality.

The matrix inequality problem (3.6) and output feedback control are closely related. Consider the system (3.3) and fixed order controller (3.4). Let us define

$$\begin{aligned}
\bar{x} &= [x^T \;\; x_c^T]^T \\
\bar{A} &= \begin{bmatrix} A & 0_{n\times n_c} \\ 0_{n_c\times n} & 0_{n_c\times n_c} \end{bmatrix}, \quad \bar{B}_1 = \begin{bmatrix} B \\ 0_{n_c\times m} \end{bmatrix}, \quad \bar{B}_2 = \begin{bmatrix} B_2 & 0_{n\times n_c} \\ 0_{n_c\times m} & I_{n_c} \end{bmatrix}, \\
\bar{C}_1 &= [C_1 \;\; 0_{r\times n_c}], \quad \bar{C}_2 = \begin{bmatrix} C_2 & 0_{r\times n_c} \\ 0_{n_c\times n} & I_{n_c} \end{bmatrix}, \\
\bar{D}_{12} &= [D_{12} \;\; 0_{r\times n_c}], \quad \bar{D}_{21} = \begin{bmatrix} D_{21} \\ 0_{r\times m} \end{bmatrix}, \quad \bar{D}_{11} = D_{11} \\
\bar{K} &= \begin{bmatrix} D_c & C_c \\ B_c & A_c \end{bmatrix}
\end{aligned} \tag{3.7}$$

Then, the closed-loop system can be written as

$$\begin{aligned}
\dot{\bar{x}}(t) &= (\bar{A} + \bar{B}_2\bar{K}\bar{C}_2)\bar{x}(t) + (\bar{B}_1 + \bar{B}_2\bar{K}\bar{D}_{21})w(t) \\
z(t) &= (\bar{C}_1 + \bar{D}_{21}\bar{K}\bar{C}_2)\bar{x}(t) + (\bar{D}_{11} + \bar{D}_{12}\bar{K}\bar{D}_{21})w(t)
\end{aligned} \tag{3.8}$$

The following results are known; see [16] and [18].

Lemma 3.1 *The system (3.3) admits an output feedback stabilizer of order n_c if and only if there exist positive definite matrices $X, Y \in R_s^{n+n_c}$ such that*

$$\begin{aligned}
L_1(X) &:= (\bar{C}_2)_\perp^T(\bar{A}^T X + X\bar{A})(\bar{C}_2)_\perp < 0 \\
L_2(Y) &:= (\bar{B}_2^T)_\perp^T(\bar{A}Y + Y\bar{A}^T)(\bar{B}_2^T)_\perp < 0 \\
XY &= I
\end{aligned} \tag{3.9}$$

In the above, the notation $U_\perp$ for a given matrix U denotes any full rank matrix whose columns form the basis of the null space of U, in particular, $UU_\perp = I$.

Lemma 3.2 *Given the system (3.3), there exists an output feedback stabilizer of order n_c such that the closed-loop transfer function has H_∞ norm less than 1 if and only if there exist positive definite matrices $X, Y \in R_s^{n+n_c}$ such that*

$$\begin{aligned}
L_1(X) &:= \begin{bmatrix} \bar{C}_2 \\ \bar{D}_{21} \\ 0 \end{bmatrix}_\perp^T \begin{bmatrix} \bar{A}^T X + X\bar{A} & X\bar{B}_1 & \bar{C}_1^T \\ \bar{B}_1^T X & -I & 0 \\ \bar{C}_1 & 0 & -I \end{bmatrix} [\bar{C}_2 \quad \bar{D}_{21} \quad 0]_\perp < 0 \\
L_2(Y) &:= [\bar{B}_2^T \quad 0 \quad \bar{D}_{21}^T]_\perp^T \begin{bmatrix} \bar{A}Y + Y\bar{A}^T & \bar{B}_1 & Y\bar{C}_1^T \\ \bar{B}_1^T & -I & 0 \\ \bar{C}_1 Y & 0 & -I \end{bmatrix} [\bar{B}_2^T \quad 0 \quad \bar{D}_{21}^T]_\perp < 0 \\
XY &= I
\end{aligned} \tag{3.10}$$

We see in Lemmas 3.1-3.2 that both cases yield the matrix inequality problem (3.6). Heuristic algorithms are commonly used in solving bilinear matrix inequalities. For example, the following iterative algorithm was proposed in [17]:

$$\begin{aligned}
X_k &= \text{argmin}\{\alpha : L_1(X) < 0, \ I \le Y_k^{1/2} X Y_k^{1/2} \le \alpha I\}, \\
Y_{k+1} &= \text{argmin}\{\beta : L_2(Y) < 0, \ I \le X_k^{1/2} Y X_k^{1/2} \le \beta I\}, \quad k = 0, 1, \ldots
\end{aligned} \tag{3.11}$$

with some initial positive definite Y_0 satisfying $L_2(Y_0) < 0$. However, such algorithms do not lead to satisfactory solutions in general.

The question we ask is whether there exist numerically tractable algorithms for solving the matrix inequality problem (3.6). The answer is, unfortunately, negative, as shown in Fu and Luo [14]:

Theorem 3.2 *The matrix inequality problem (3.6) is NP-hard.*

3.3 Pole Placement Approach

We now consider the problem of pole-placement using static output feedback. Given the system (3.1) and a set of desired eigenvalues $\lambda_i, i = 1, 2, \ldots, \nu \le n$, we want to know whether there exists static output feedback controller (3.2) such that all λ_i are the closed-loop eigenvalues.

Denoting the open-loop transfer function by $G(s) = C(sI - A)^{-1}B$, the pole placement problem is equivalent to finding K such that

$$\det(I + KG(\lambda_i)) = 0, i = 1, 2, \ldots, q \tag{3.12}$$

Again, from computational complexity point of view, we want to know if there exists an efficient algorithm to determine the existence of such K. Unfortunately, we have a negative result for this problem [9]:

Theorem 3.3 *The static output feedback pole placement problem is NP-hard.*

Remark: It is obvious that if there is a sufficient number of independent outputs, pole placement is always possible. State feedback pole placement is a special case of this fact. Therefore, the result above can be interpreted as that determining whether there is a sufficient number of outputs for pole placement is NP hard.

4 The Structured Singular Value (μ) Problem

The structured singular value problem, also known as the μ problem, is the central problem in robustness analysis for control systems. Although it is mostly studied in the control field, its implications and applications are far reaching because it addresses a fundamental numerical analysis and linear algebraic problem.

The term of structured singular value was coined by Doyle [6], although its concept had been around for a long time for robustness analysis of linear systems subject to various types of uncertainties. Three types of uncertainties are considered: real parameters, complex parameters and complex blocks. Real parameters correspond to common physical parameters such as mass, friction coefficient, resistance, etc., and are described by a block structure below:

$$\mathbf{\Delta}_r = \{\Delta_r = \mathrm{diag}\{\delta_1 I_{r_1}, \delta_2 I_{r_2}, \ldots, \delta_m I_{r_m}\} | \delta_i \in \mathbb{R}\} \tag{4.1}$$

Complex parameters are less common, but useful for describing uncertainties in complex-valued signals such as modulated radio waveforms, and are described by

$$\mathbf{\Delta}_c = \{\Delta_c = \mathrm{diag}\{z_1 I_{c_1}, z_2 I_{c_2}, \ldots, z_t I_{c_t}\} | z_i \in \mathbb{C}\} \tag{4.2}$$

Complex block uncertainties are typically used to model unstructured uncertainties (e.g., unmodeled dynamics in transfer functions), and are described by

$$\mathbf{\Delta}_C = \{\Delta_C = \mathrm{diag}\{\Delta_1, \Delta_2, \ldots, \Delta_v\} | \Delta_i \in \mathbb{C}^{k_i \times k_i}\} \tag{4.3}$$

When different types of uncertainties are present, we have the so-called mixed uncertainties.

The general structured singular value problem, also known as the mixed μ problem, is described as follows: Given a complex matrix $M \in \mathbb{C}^{n\times n}$ and a set $\mathbf{\Delta}$ defined by

$$\mathbf{\Delta} = \{\Delta = \mathrm{diag}\{\Delta_r, \Delta_c, \Delta_C\} | \Delta_r \in \mathbf{\Delta}_r, \Delta_c \in \mathbf{\Delta}_c, \Delta_C \in \mathbf{\Delta}_C\} \tag{4.4}$$

with

$$r_1 + r_2 + \ldots + r_m + c_1 + c_2 + \ldots + c_t + k_1 + k_2 + \ldots + k_m = n \tag{4.5}$$

The problem of structured singular value analysis is to compute the value of the function $\mu_{\mathbf{\Delta}}(M)$, which is defined to be zero if $I_n - \Delta M$ is nonsingular for all $\Delta \in \mathbf{\Delta}$, or otherwise

$$\mu_{\mathbf{\Delta}} = \frac{1}{\inf\{\rho > 0 | \det(\rho I_n - \Delta M) = 0,\ \Delta \in \mathbf{B}(\mathbf{\Delta})\}} \tag{4.6}$$

where

$$\mathbf{B}(\mathbf{\Delta}) = \{\Delta \in \mathbf{\Delta} |\ |\delta_i| \leq 1,\ |z_j| \leq 1,\ \|\Delta_k\| \leq 1,\ \forall i, j, k\} \tag{4.7}$$

Two special cases are important: The real μ problem where only $\mathbf{\Delta}_r$ is present; and the pure complex μ (or complex μ for short) problem where only $\mathbf{\Delta}_C$ is present.

4.1 Existing algorithms

The first algorithm for computing the pure complex μ value was proposed by Doyle [6], and it is known as the D-scaling method. The idea is as follows: Suppose the uncertainty structure $\mathbf{\Delta}$ involves complex blocks only, i.e.,

$$\mathbf{\Delta} = \{\mathrm{diag}\{\Delta_1, \Delta_2, \ldots, \Delta_v\} | \Delta_i \in \mathbb{C}^{k_i \times k_i}\} \tag{4.8}$$

Let

$$\mathbf{D} = \{D = \text{diag}\{d_1 I_{k_1}, d_2 I_{k_2}, \ldots, d_v I_{k_v}\} | d_i \in \mathbb{P}\} \tag{4.9}$$

where $\mathbb{P}$ denotes the set of positive real numbers. Then, for any $\Delta \in \mathbf{B}(\boldsymbol{\Delta})$, $D \in \mathbf{D}$ and $\rho > 0$, $\rho I - \Delta M$ is nonsingular if and only if $\rho I - \Delta D^{1/2} M D^{-1/2}$ is nonsingular because $D^{1/2}$ and Δ commute. A sufficient condition for $\rho I - \Delta D^{1/2} M D^{-1/2}$ to be nonsingular for all $\Delta \in \mathbf{B}(\boldsymbol{\Delta})$ is that $\|D^{1/2} M D^{-1/2}\| < \rho$, which is equivalent to $M^* D M < \rho^2 D$. Changing ρ^2 to α and defining

$$\Phi_\alpha(D) = M^* D M - \alpha D \tag{4.10}$$

the following provides an upper bound for $\mu_{\boldsymbol{\Delta}}(M)$:

$$\begin{array}{ll} \min & \alpha \\ \text{subject to} & \Phi_\alpha(D) < 0, D \in \mathbf{D}, \alpha > 0 \end{array} \tag{4.11}$$

The above is an example of the so-called generalized eigenvalue problem, which can be solved using semidefinite programming (a polynomial time algorithm) [2].

The D-scaling method was generalized in [8] to the so-called (D,G)-scaling method for the mixed μ problem. This is done as follows: Given the structure $\boldsymbol{\Delta}$ in (4.4), let

$$\begin{aligned} \mathbf{D} = \{ & D = \text{diag}\{P_{r_1}, P_{r_2}, \ldots, P_{r_m}, P_{c1}, P_{c2}, \ldots, P_{c_t}, \\ & d_1 I_{k_1}, d_2 I_{k_2}, \ldots, d_v I_{k_v}\} | P_{r_i} \in \mathbb{P}^{r_i}, P_{c_j} \in \mathbb{P}^{c_j}, d_k \in \mathbb{P}\} \end{aligned} \tag{4.12}$$

where $\mathbb{P}^k$ denotes the set of $k \times k$ positive definite matrices, and let

$$\begin{aligned} \mathbf{G} = \{ & G = \text{diag}\{H_{r_1}, H_{r_2}, \ldots, H_{r_m}, 0_{c1}, 0_{c2}, \ldots, 0_{c_t}, \\ & 0_{k_1}, 0_{k_2}, \ldots, 0_{k_v}\} | H_{r_i} \in \mathbb{H}^{r_i}\} \end{aligned} \tag{4.13}$$

where $\mathbb{H}^k$ denotes the set of $k \times k$ hermitian matrices. Then, for any $\Delta \in \mathbf{B}(\boldsymbol{\Delta})$, $D \in \mathbf{D}$ and $G \in \mathbf{G}$, Δ commute with both D and G. Define

$$\Phi_\alpha(D, G) = M^* D M - \alpha D + j(GM - M^* G), \ D \in \mathbf{D}, \ G \in \mathbf{G} \tag{4.14}$$

Then, like the pure complex μ case, an upper bound for $\mu_{\boldsymbol{\Delta}}(M)$ is given by

$$\begin{array}{ll} \min & \alpha \\ \text{subject to} & \Phi_\alpha(D, G) < 0, D \in \mathbf{D}, D \in \mathbf{G}, \alpha > 0 \end{array} \tag{4.15}$$

The (D,G)-scaling method has been the dominant method for μ analysis. For this reason, a lot of research has been devoted to studying the quality of this method; see, e.g., [22, 20].

One line of research focuses on conditions under which the upper bound is exact, i.e., the upper bound gives the exact μ value. A complete result is given in a paper by Meinsma, Shrivastava and Fu [21], which shows that the number of cases where the upper bound is exact is rather limited. Recalling that the numbers of real parameters, complex parameters and complex blocks are denoted by m, t and v, respectively. The exact cases are

- $m = 0, t = 0, v \leq 3$;

- $m = 0, t = 1, v \leq 1$;
- $m = 1, t = 0, v \leq 1$.

All other cases have examples to show that the upper bound is not exact.

Another line of research aims to providing tighter (i.e., better) upper bounds. In Fu and Barabanov [12], a number of new upper bounds are introduced based on the so-called multiplier approach which is simply stated as follows:

Lemma 4.1 *[12] Given a complex matrix $M \in \mathbb{C}^n$, an uncertainty structure $\boldsymbol{\Delta}$ as in (4.4) and $\rho > 0$, $\rho I - \Delta M$ is nonsingular for all $\Delta \in \mathbf{B}(\boldsymbol{\Delta})$ if there exists a matrix $C \in \mathbb{C}^n$ such that*

$$C(\rho I - \Delta M) + (\rho I - \Delta M)^* C^* < 0, \quad \forall \Delta \in \mathbf{B}(\boldsymbol{\Delta}) \tag{4.16}$$

The parameter C above is called multiplier. A nice property of the result above is that the condition (4.16) is linear in Δ. This result provides a simple way to obtain an upper bound for μ. Namely, the upper bound for $\mu_{\boldsymbol{\Delta}}(M)$ is the reciprocal of the least value of ρ for which a multiplier C exists for (4.16). This approach is conceptually simpler than the (D,G)-scaling method, but can be shown to include (D,G)-scaling as a special case [12, 20]. Algorithms based on the multiplier approach are in general tighter than (D,G)-scaling, but at the price of more computational time; see [12] for more details.

4.2 Computational Complexity

The real μ problem was shown to be NP-hard first by Poljak and Rohn [24]. A similar result was given by Demmel [5]. Their proofs used the idea of transforming the so-called max-cut problem, a known NP-complete problem into a real μ problem in polynomial time. Since the real μ problem is a special case a mixed μ problem, these results imply that a general μ problem is also NP-hard.

In the works of Braatz *et. al.* [3] and Coxson and DeMarco [4], a different NP-complete problem, minimization of concave quadratic function, is used to prove the NP-hardness of the mixed μ problem. Their approaches are algebraic and simpler. Also shown in [3] is that adding complex uncertainties does not make the μ problem easier.

In all the proofs above, real uncertainties are essential in showing the NP-hardness. The computational complexity of the complex μ problem was also shown to be NP-hard by Toker and Ozbay [26] using an elegant transformation from another NP-complete problem.

Knowing that the problem of computing μ is NP-hard in various cases, the next logical question is how "hard" it is to approximate μ. To this end, a result in [4] shows that there exists some arbitrarily small $\varepsilon > 0$ such that ε-approximation for the real μ is also NP-hard. Toker [25] offers a more negative answer for the real μ problem by showing that computing a $Cn^{1-\varepsilon}$-approximation with some (very large) $C > 0$ and (very small) $\varepsilon > 0$ is also an NP-hard problem. In the above, n refers to the dimension of the μ problem, and $Cn^{1-\varepsilon}$-approximation means that the relative approximation error is sublinear, i.e., it does not grow linearly as the dimension n grows. So the NP-hardness of the $Cn^{1-\varepsilon}$-approximation problem suggests that no polynomial algorithms can give approximation of μ with relative error growing at any sublinear rate, unless P=NP.

In Fu [10], a most negative result was given for the real μ problem: The problem of $r(n)$-approximation for the real μ is NP-hard for any $r(n) > 0$. That is, the relative approximation error can grow arbitrarily fast for any polynomial algorithm, unless P=NP. The result in [10] is further extended by Fu and Dasgupta [13] to the case where the real blocks are bounded using any p-norm rather than the ∞-norm as in the standard case.

Although the computational complexity of the problem of approximating the real μ value is well understood, the case for complex μ is less clear. The only known result was given by Megretski [19] which shows that the D-scaling method gives a relative error r which grows at most linearly as the function of n, provided that $\mu \neq 0$. That is, the problem of finding a linearly growing r-approximation for the complex μ is a polynomial problem, provided $\mu \neq 0$.

It was proposed as an open problem by Fu in [11] to determine whether the problem of approximating complex μ with an arbitrarily small or arbitrarily large *constant* relative error is NP-hard. It was conjectured in [11] that both cases are NP-hard. But the answers are still unknown.

5 Conclusions

This chapter has introduced two fundamental yet challenging problems in the control theory. The static output feedback stabilization appears to be a simple problem, yet so far it is not known whether the problem is NP-hard or not. The main challenge is to determine if this is the case or not. Solving this problem is important to a wide range of fixed-order control problems. On the other hand, better heuristic algorithms are still needed so that practical control designs can be carried out better. The structured singular value problem remains a major problem in robust control, despite the fact that many NP-hard results are known. One particular challenge is to determine how well we can approximate complex μ. Also, better heuristic algorithms, e.g., algorithms faster but giving tighter bounds than (D,G)-scaling, will be very valuable to robustness analysis and robust control design.

References

[1] V. Blondel and J. N. Tsitsiklis, “NP-hardness of some linear control design problems,” *SIAM J. Control Optim.*, vol. 35, no. 6, pp. 2118-2127, Nov. 1997.

[2] S. Boyd, *et. al.*, *Linear Matrix Inequlaties in Systems and Control Theory*, Philadelphia, PA: SIAM, 1994.

[3] R. P. Braatz *et. al.*, “Computational complexity of μ calculation,” *IEEE Trans. Automat. Contr.*, vol. 39, no. 5, pp. 1000-1002, May 1994.

[4] G. E. Coxson and C. L. DeMarco, “The computational complexity of approximating the minimal perturbation scaling to achieve instability in an interval matrix,” *Math. Contr. Signals Syst.*, vol. 7, no. 4, pp. 279-292, 1994.

[5] J. W. Demmel, "The component-wise distance to the nearest singular matrix," *SIAM J. Matrix Anal. Appl.*, vol. 13, pp. 10-19, 1992.

[6] J. C. Doyle, "Analysis of feedback systems with sructured uncertainties," *Proc. Inst. Elec. Eng.*, vol. 129, pt. D., pp. 240-250, 1982.

[7] J. C. Doyle, K. Glover, P. P. Khargonekar and B. A. Fracis, "State-space solutions to the standard H^2 and H^∞ control problems," *IEEE Trans. Automat. Contr.*, vol. 34, no. 8, pp. 831-847, 1989.

[8] M. K. H. Fan, A. L. Tits and J. C. Doyle, "Robustness in the presence of mixed parametric uncertainty and unmodelled dynamics," *IEEE Trans. Automat. Contr.*, vol. 36, no. 1, pp. 25-38, Jan. 1991.

[9] M. Fu, "Pole placement via static output feedback is NP-hard," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 855-857, 2004.

[10] M. Fu, "The Real μ is Hardly Approximable," *IEEE Trans. Auto. Contr.*, vol. 42, no. 9, pp. 1286-1288, 1997.

[11] M. Fu, "Approximation of complex μ," in *Open Problems in Mathematical Systems Theory*, eds. V. Blondel, E. Sontag, M. Vidyasagar and J. Willems, Springer Verlag, 1998.

[12] M. Fu and N. E. Barabanov, "Improved upper bounds of the mixed structured singular value," *IEEE Trans. Auto. Contr.*, vol. 42, no. 10, pp. 1447-1452, 1997.

[13] M. Fu and S. Dasgupta, "Computational complexity of real structured singular value in l_p setting," *IEEE Trans. Automat. Contr.*, vol. 45, no. 11. pp. 2173 -2176, 2000.

[14] M. Fu and Z.-Q. Luo, "Computational complexity of a problem arising in fixed order output feedback design," *Systems and Control Letters*, vol. 30, pp. 209-215, 1997.

[15] M. R. Gary and D. S. Johnson, *Computers and Intractability: A Guide to NP-Completeness.* New York: W. H. Freeman, 1983.

[16] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to H_∞ control," *Int. J. Robust and Nonlinear Control*, vol. 4, pp. 421-448, 1992.

[17] J. C. Geromel, P. L. D. Peres and R. E. Skelton, "LMI numerical solution for output feedback stabilization," *Proc. American Control Conf.*, Baltimore, MD, 1994.

[18] T. Iwasaki and R. E. Skelton, "The XY-centering algorithm for the dual LMI problem: A new approach to fixed order control design," *Int. J. Control*, vol. 62, pp. 1257-1272.

[19] A. Megretski, "On the gap between structured singular values and their upper bounds," *Proc. 32nd Conf. Decision and Control*, San Antonio, TX, Dec. 1993.

[20] G. Meinsma, Y. Shrivastava and M. Fu, "Some Properties of an Upper Bound for μ," *IEEE Trans. Auto. Contr.*, vol. 41, no. 9, pp. 1326-1330, 1996.

[21] G. Meinsma, Y. Shrivastava and M. Fu, "A Dual Formulation of Mixed mu and the Losslessness of (D,G)-scaling," *IEEE Trans. on Auto. Contr.*, vol. 42, no. 7, pp. 1032-1036, 1997.

[22] A. Packard and J. Doyle, "The complex structured singular value," *Automatica*, vol. 29, no. 1, pp. 71-109, 1993.

[23] C. H. Papadimitriou and K. Steglitz, *Combinatorial Optimization: Algorithms and Complexity.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[24] S. Poljak and J. Rohn, "Checking robust nonsingularity is NP-hard," *Math. Contr., Signals, Syst.*, vol. 6, pp. 1-9, 1993.

[25] O. Toker, "On the conservatism of upper bound tests for structured singular value analysis," *Proc. 35th Conf. Decision and Contr.*, Kobe, Japan, Dec. 1996.

[26] O. Toker and H. Ozbay, "On the complexity of purely complex μ computation and related problems in multidimensional systems," *IEEE Trans. Auto. Contr.*, vol. 43, no. 3, pp. 409-414, 1998.

[27] W. M. Wonham, *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, New York, 1979.