Journal of Computational Mathematics, Vol.25, No.6, 2007, 672–689.

FIRST-ORDER METHODS FOR SOLVING THE OPTIMAL STATIC $\mathcal{H}_\infty$-SYNTHESIS PROBLEM *

El-Sayed M.E. Mostafa
(*Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt*
Email: emostafa99@yahoo.com)

Abstract

In this paper, we consider the static output feedback (SOF) $\mathcal{H}_\infty$-synthesis problem posed as a nonlinear semi-definite programming (NSDP) problem. Two numerical algorithms are developed to tackle the NSDP problem by solving the corresponding Karush-Kuhn-Tucker first-order necessary optimality conditions iteratively. Numerical results for various benchmark problems illustrating the performance of the proposed methods are given.

Mathematics subject classification: 49N35, 49N10, 93D52, 93D22, 65K05.
Key words: Static output feedback, $\mathcal{H}_\infty$-synthesis, Semi-definite programming, Nonlinear programming.

1. Introduction

In this paper, we consider the following NSDP problem:

$$\text{(P0)}: \begin{cases} \min\limits_{X,V,\gamma} & \gamma \\ \text{s.t.} & H(X,\gamma)=0, \quad Y(X,V,\gamma)\prec 0, \quad V\succ 0, \end{cases} \tag{1.1}$$

where $\gamma \in \mathbb{R}_+$, and $H: \mathbb{R}^{r\times t}\times \mathcal{S}^n \times \mathbb{R}_+ \to \mathcal{S}^n$, $Y: \mathbb{R}^{r\times t}\times \mathcal{S}^n \times \mathcal{S}^n \times \mathbb{R}_+ \to \mathcal{S}^n$ are assumed to be sufficiently smooth matrix functions. In the considered optimal control applications the variable X is a decomposition of a matrix pair $(F,L)\in \mathbb{R}^{r\times t}\times \mathcal{S}^n$, where $\mathcal{S}^n$ denotes the set of real symmetric $n\times n$ matrices. The problem (1.1) is a nonlinear semi-definite programming and is generally non-convex.

In recent years there were several attempts to employ the available successful computational techniques in nonlinear optimization to solve various NSDP problems numerically. Such NSDP problems represent a variety of applications in system and control theory; see among others [6, 7, 9, 12, 14, 17, 18, 19, 22, 24, 32]. In particular, the above NSDP problem (1.1) represents a wide range of applications in system and control theory; see, e.g., the benchmark collection COMPl$_\text{e}$ib [25].

By solving (1.1) we mean computing a feasible point (X,V,γ) that satisfies the set of equality and inequality constraints as well as enforcing the objective γ to attain its least possible value. Due to the difficulties in solving problem (1.1) numerically, however, in most of the above citations the attempts were only to compute suboptimal solution.

The main goal of this paper is to propose two first-order methods for solving (1.1) that take advantage of the problem structure and use inexact computations. In these methods we

* Received July 30, 2005; final revised March 8, 2006; accepted June 29, 2006.

compute a stationary point of the above optimization problem by solving numerically the nonlinear system of equations resulting from the Karush-Kuhn-Tucker (KKT) first-order necessary optimality conditions. It is particularly important in the proposed methods to deal with a scalar objective function together with matrix variables.

This paper is organized as follows. In the next section we present the problem formulation and then we state the assumptions imposed on the problem (1.1). In addition, we obtain the nonlinear system of equations resulting from the first-order necessary optimality conditions of the problem (1.1). In Section 3 we develop two first-order methods for computing approximately a stationary point of the problem (1.1). In Section 4 we test numerically the proposed algorithms through several test problems from the benchmark collection COMPl$_\text{e}$ib [25].

Notations: For a matrix $M \in \mathbb{R}^{n\times n}$ the notations $M \succ 0$, $M \prec 0$ denote that M is positive definite, negative definite, respectively. Throughout the paper, the symbol $\|\cdot\|$ denotes the Frobenius norm defined by $\|M\| = \sqrt{\langle M, M\rangle}$, where $\langle\cdot,\cdot\rangle$ is the inner product given by $\langle M_1, M_2\rangle = Tr\,(M_1^T M_2)$, and $Tr\,(\cdot)$ is the trace operator.

2. Problem Formulation

The solution of the $\mathcal{H}_\infty$ synthesis problem has received considerable attention in the control literature; see, e.g., [1-3, 6-10, 12, 14, 15, 17-19, 22, 24, 31, 32] and the references therein. Given a linear time-invariant (LTI) control system, the $\mathcal{H}_\infty$ synthesis problem can be stated as follows: Find an output feedback control matrix F that minimizes the $\mathcal{H}_\infty$ norm of a certain transfer function subject to the constraint that this control matrix F is stabilizing the associated control system. A typical instance of an output feedback control system can be stated as follows. Consider the LTI control system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + B_1 w(t), \\ y(t) &= Cx(t), \\ z(t) &= C_1 x(t) + D_1 u(t), \end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}^{n_x}$, $w \in \mathbb{R}^{n_w}$, $u \in \mathbb{R}^{n_u}$, $z \in \mathbb{R}^{n_z}$, and $y \in \mathbb{R}^{n_y}$ denote the state, the disturbance input, the control input, the regulated output, and the measured output, respectively. Furthermore, A, B_1, B, C_1, C, and D_1 are given constant matrices of appropriate dimensions.

We consider the static output feedback (SOF) control law:

$$u(t) = Fy(t), \tag{2.2}$$

where $F \in \mathbb{R}^{n_u\times n_y}$ denotes the unknown static output feedback gain, which we attempt to compute by a suitable numerical procedure.

Given an output feedback matrix F and a control system (2.1), the closed-loop counterpart is given by:

$$\begin{aligned} \dot{x}(t) &= A(F)x(t) + B(F)w(t), \\ z(t) &= C(F)x(t), \end{aligned} \tag{2.3}$$

where

$$A(F) := A + BFC, \quad B(F) := B_1, \quad C(F) := C_1 + D_1 FC$$

are the augmented closed loop operators, respectively.

The following assumption is needed throughout the paper.

Assumption 2.1. *The matrix $C \in \mathbb{R}^{n_y \times n_x}$ has full row rank, and the matrix product $D_1^T D_1$ is invertible.*

The optimal SOF $\mathcal{H}_\infty$-synthesis problem (see the above citations) is equivalent to the following NSDP problem:

$$\text{(P)} : \begin{cases} \min\limits_{F,\,L,\,\gamma} & \gamma \\ \text{s.t.} & H(F,L;\gamma) = 0, \quad Y(F,L;\gamma) \prec 0, \quad L \succ 0, \end{cases} \tag{2.4}$$

where $H : \mathbb{R}^{n_u \times n_y} \times \mathcal{S}^n \times \mathbb{R} \to \mathcal{S}^n$, $Y : \mathbb{R}^{n_u \times n_y} \times \mathcal{S}^n \times \mathbb{R} \to \mathcal{S}^n$ are defined by

$$H(F,L;\gamma) = A(F)^T L + LA(F) + C(F)^T C(F) + \frac{1}{\gamma^2} L B_1 B_1^T L,$$

$$Y(F,L;\gamma) = (A(F) + \frac{1}{\gamma^2} B_1 B_1^T L)^T L + L(A(F) + \frac{1}{\gamma^2} B_1 B_1^T L),$$

which are sufficiently smooth matrix functions, $\gamma \in \mathbb{R}_+$, and $\mathcal{S}^n$ denotes the set of real symmetric $n \times n$ matrices. In the optimal control terminology F represents the control, L represents the state, the equality constraint represents the state equation, and the coupled inequality constraints replace the (asymptotic) stability constraint in the Lyapunov sense for which the spectral abscissa of the matrix $A(F)$ defined by

$$\alpha(A(F)) = \max_i \operatorname{Re}(\lambda_i(A(F))), \tag{2.5}$$

must be strictly negative; in other words, the largest real-part of the eigenvalues of $A(F)$ must be strictly negative.

The Lagrangian function associated with the equality constraint of (2.4) is defined by

$$\mathcal{L}(F,L,K,\gamma) = \gamma + \langle K, H(F,L,\gamma) \rangle, \tag{2.6}$$

where $K \in \mathbb{R}^{n \times n}$ is the associated Lagrange multiplier.

The formulation of the NSDP problem (2.4) is clearly involving bilinear matrix inequalities (BMIs). The $\mathcal{H}_\infty$-synthesis problem using BMIs has been considered, e.g., in [10, 20, 22, 26]. Typically through this formulation the optimal static $\mathcal{H}_\infty$-synthesis problem can be treated as a regular nonlinear programming problem, where various available successful numerical techniques can be employed.

For solving the problem (2.4) we have two different view points regarding γ. First we consider γ as a variable. In this case we can differentiate $\mathcal{L}$ with respect to γ, and therefore γ enters into the search direction. In the second case we consider γ to be a parameter. Consequently, we do not differentiate $\mathcal{L}$ with respect to γ and as a result we try to estimate γ using one of the available equations of the first-order necessary optimality conditions. Two numerical algorithms are developed corresponding to the two cases. Although only first-order information are involved the numerical results given in Section 4 show that this approach is quite successful for computing a stationary point to the problem (2.4).

Assumption 2.2. *The following assumptions are used throughout the paper:*

- **AS1.** *There exists an initial $(F_0, L_0, \gamma_0) \in \mathcal{F}_s$, where*

$$\mathcal{F}_s = \Big\{ (F,L,\gamma) \in \mathbb{R}^{n_u \times n_y} \times \mathcal{S}^n \times \mathbb{R}_+ \mid Y(F,L,\gamma) \prec 0, \quad L \succ 0 \Big\}.$$

- **AS2.** *There exists a* $(F_*, L_*, \gamma_*) \in \mathcal{F}_s$ *solution of the NSDP problem* (2.4).

Next, let us compute the first-order directional derivatives of the Lagrangian function (2.6), which are required in the derivation of the computational methods.

Lemma 2.1. *Let* $(F, L, \gamma) \in \mathcal{F}_s$ *and* $K \in \mathbb{R}^{n_x \times n_x}$ *be given. Then, the constraint function* H *is continuously differentiable on* $\mathcal{F}_s$. *Furthermore, the first-order directional derivatives of the Lagrangian function in the direction* $(\Delta F, \Delta L, \Delta\gamma)$ *are given by*

$$\begin{aligned}
&\mathcal{L}_F(F, L, K, \gamma)\Delta F \equiv \langle \Delta F, \nabla_F \mathcal{L}(\cdot)\rangle = \langle \Delta F, 2(B^T L + D_1^T C(F))KC^T\rangle,\\
&\mathcal{L}_L(F, L, K, \gamma)\Delta L \equiv \langle \Delta L, \nabla_L \mathcal{L}(\cdot)\rangle = \langle \Delta L, \tilde{A}(\cdot)K + K\tilde{A}(\cdot)^T\rangle,\\
&\mathcal{L}_K(F, L, K, \gamma)\Delta K \equiv \langle \Delta K, \nabla_K \mathcal{L}(\cdot)\rangle = \langle \Delta K, H(F, L, \gamma)\rangle,\\
&\mathcal{L}_\gamma(F, L, K, \gamma)\Delta\gamma \equiv \Big[1 - \frac{2}{\gamma^3}\langle K, LB_1B_1^T L\rangle\Big]\Delta\gamma,
\end{aligned}$$

where

$$\tilde{A}(\cdot) \equiv \tilde{A}(F, L, \gamma) = (A(F) + \frac{1}{\gamma^2}B_1B_1^T L).$$

Proof. The first-order directional derivatives of $\mathcal{L}(F, L, K, \gamma)$ with respect to F, L, K, and γ in the direction of $(\Delta F, \Delta L, \Delta K, \Delta\gamma)$ yield the above derivatives, where the trace properties are used.

The KKT necessary optimality conditions of the NSDP problem (2.4) can be obtained directly from the result of Lemma 2.1, which are the following:

$$(B^T L + D_1^T C(F))KC^T = 0, \tag{2.7}$$

$$(A(F) + \frac{1}{\gamma^2}B_1B_1^T L)K + K(A(F) + \frac{1}{\gamma^2}B_1B_1^T L)^T = 0, \tag{2.8}$$

$$2\,\langle K,\, LB_1B_1^T L\rangle = \gamma^3, \tag{2.9}$$

$$LA(F) + A(F)^T L + C(F)^T C(F) + \frac{1}{\gamma^2}LB_1B_1^T L = 0. \tag{2.10}$$

For optimal control problems (2.10) is the state equation, (2.8) corresponds to the adjoint equation, while the left-hand side of (2.7) is the gradient. Obviously, the explicit analytical expression for γ given by (2.9) can be used for estimating such a scalar variable for given K and L. On the other hand, (2.8) can be rewritten as a Lyapunov equation of the form

$$A(F)K + K\,A(F)^T + \frac{1}{\gamma^2}(B_1B_1^T LK + KLB_1B_1^T) = 0. \tag{2.11}$$

3. First-order Methods for Solving the Problem P

In the next subsection, a first-order method is proposed for computing approximately a stationary point of the NSDP problem (2.4) by solving the KKT system (2.7)-(2.10) iteratively. This method can be considered as a Lagrangian method and is pioneered by the algorithm of Levine and Athans [23].

3.1. Treating γ as a variable

The attempt is to solve the KKT system (2.7)-(2.10) iteratively using only first-order information. The procedure is described as follows. From the Assumption 2.2 and if K is positive definite, the matrix inverse $(CKC^T)^{-1}$ exists and (2.7) implies

$$\tilde{F}(L, K) = -(D_1^T D_1)^{-1}(B^T L + D_1^T C_1)KC^T(CKC^T)^{-1}. \tag{3.1}$$

Then, given the new estimate $\tilde{F}$ we solve the Lyapunov equation (2.10):

$$LA(\tilde{F}) + A(\tilde{F})^T L + \left[C(\tilde{F})^T C(\tilde{F}) + \frac{1}{\gamma^2} LB_1 B_1^T L\right] = 0. \tag{3.2}$$

Let $\tilde{L}$ be the corresponding solution. Next, a new estimate for γ can be obtained explicitly by using (2.9):

$$\tilde{\gamma} := \gamma(\tilde{L}, K) = \sqrt[3]{2\,\langle K,\, \tilde{L}B_1 B_1^T \tilde{L}\rangle}. \tag{3.3}$$

The final step is to plug the nonlinear term $\tilde{F}(L, K)$ of (3.1) and the computed $\tilde{\gamma}$ into (2.8). It implies the matrix equation

$$\begin{aligned}
&\Big[A - B(D_1^T D_1)^{-1}(B^T L + D_1^T C_1)KC^T(CKC^T)^{-1}C\Big]K \\
&\quad + K\Big[A - B(D_1^T D_1)^{-1}(B^T L + D_1^T C_1)KC^T(CKC^T)^{-1}C\Big]^T \\
&\quad + \frac{1}{\tilde{\gamma}^2}\Big(B_1 B_1^T \tilde{L}K + K\tilde{L}B_1 B_1^T\Big) = 0,
\end{aligned} \tag{3.4}$$

which is nonlinear in the unknown K and can be solved by any nonlinear solver. Let $\tilde{K}$ denote the computed solution.

From Lyapunov stability theory it is well known (see, e.g., [5]) that the closed-loop control system (2.3) is asymptotically stable if and only if there exists a triplet $(\tilde{F}, \tilde{L}, \tilde{\gamma})$ such that

$$\tilde{L} \succ 0 \quad \text{and} \quad Y(\tilde{F}, \tilde{L}, \tilde{\gamma}) \prec 0. \tag{3.5}$$

Hence, it is also possible to replace the computation of the spectral abscissa $\alpha(A(\tilde{F}))$ in every iteration by checking the positive/negative definiteness constraints (3.5). This can be done efficiently by using the incomplete Cholesky decomposition, e.g., by using the Matlab function $[R, p] =$ cholinc $(M, '0')$, where M is a square matrix. If $M \succ 0$, then p is 0; otherwise p is a positive integer. However, in the implementation we observed that this issue is not influential due to the low size of the most considered test problems; see Section 4.

It is particularly important to note that, if the computed iteration hits the boundary of the inequality constraints, then we do backtracking and stop with the corresponding least possible value of γ; see step b of the two algorithms 3.1 and 3.2 below.

A reasonable stopping criterion for this procedure is the following

$$\|\nabla_F \mathcal{L}(F, L, K, \gamma)\| + \|H(F, L, \gamma)\| \le \epsilon_{tol}, \tag{3.6}$$

where $\nabla_F \mathcal{L}(F, L, K, \gamma)$ denotes the gradient of the optimal control problem (2.4) represented by the left hand side of (2.7) and ϵ_{tol} is the tolerance. This criterion provides a measure of convergence towards a stationary point (F_*, L_*, γ_*) of the NSDP problem (2.4). Its first part measures the reduction in the gradient term, while the second part measures the fulfillment of the state equation (2.10); i.e., it measures convergence towards feasibility. The remaining parts of the KKT system (2.8)-(2.9) might be represented in the above criterion. However, (2.8) is transformed into the nonlinear equation (3.4), which depends on the used nonlinear solver. Eqn (2.9) is used for computing the new estimate of γ and consequently the difference

$$\left|\tilde{\gamma} - \sqrt[3]{2\,\langle K,\, LB_1B_1^T L\rangle}\right|$$

is negligible, and therefore that term is not included in (3.6).

We emphasize that various types of stopping criteria have been tried, among them are (3.6),

$$\max\left\{\|\nabla_F \mathcal{L}(F,L,K,\gamma)\| + \|H(F,L,\gamma)\|,\ \left|\tilde{\gamma} - \sqrt[3]{2\,\langle K,\, LB_1B_1^T L\rangle}\right|\right\} \le \epsilon_{tol},$$

and

$$|\Delta\gamma| + \|\Delta F\| + \|\Delta L\| \le \epsilon_{tol}.$$

From our numerical experience with the convergence behavior of the considered methods the stopping criterion (3.6) was the best choice; see Section 4.

The computational algorithm corresponding to the case of γ being a variable is stated in the following lines.

Algorithm 3.1. (FOM1: First-order method for solving the KKT system (2.7)-(2.10))
Given $(F_0, L_0, \gamma_0) \in \mathcal{F}_s$ with $\gamma_0 > 0$, and $K_0 \succ 0$. Choose $\beta, \epsilon \in (0,1)$, $\tau \in (0, \frac{1}{2})$.
While the termination criterion (3.6) is not reached, do

a. Given L_k and K_k compute the new estimate $F_{k+1} = \tilde{F}(L_k, K_k)$ by using (3.1).

b. Compute the spectral abscissa $\alpha(A(F_{k+1}))$.
While $\alpha(A(F_{k+1})) \ge 0$, do
(Perform backtracking)
Set $l \leftarrow l+1$ and $F_{k+1} = F_k + \beta^l\tau(F_k - F_{k+1})$.
If $\beta^l\tau \le \epsilon$, stop.

c. Given F_{k+1}, solve the Lyapunov equation (3.2) for L_{k+1}.

d. Given K_k and L_{k+1}, compute $\gamma_{k+1} = \gamma(L_{k+1}, K_k)$ by using (3.3).

e. Given γ_{k+1} and L_{k+1} solve the nonlinear matrix equation (3.4) in K_{k+1}.

f. Set $k \leftarrow k+1$ and go to step a.

end(While)

Remark 3.1. If the computed F_{k+1} in the Algorithm 3.1 is such that $\alpha(A(F_{k+1})) \ge 0$, then we do backtracking to compute the least possible γ_{k+1} and we stop if the stepsize parameter $\beta^l\tau$ reaches a small constant value $\epsilon \in (0,1)$. However, this situation is not prevailing as will be seen in the numerical results.

3.2. Treating γ as a parameter

In this subsection we consider γ as a parameter. The main attempt is to derive a computational method for solving (2.4) similar to Algorithm 3.1. As is mentioned above, the Lagrangian function is not differentiated in this case with respect to γ, and consequently (2.9) is not available any more among the KKT system (2.7)-(2.10). Therefore, the number of unknowns exceeds the number of equations. However, we can overstep this difficulty and derive a computational method for solving the reduced KKT system (2.7)-(2.8), (2.10) similar to Algorithm 3.1. In other words, we will proceed almost the same way as in the last subsection, but we estimate γ differently.

Note that in the last subsection we substitute $\tilde{F}$ into (2.11) for giving (3.4). Hence, (2.11) is not used within the method FOM1 in its given form as a Lyapunov equation. On the contrary, in the current case we use (2.11) to update γ as follows: By applying the trace operator on both sides of (2.11) followed by taking the square root, it yields

$$\gamma(F,L,K)=\sqrt{\frac{Tr\,(B_1B_1^TLK+KLB_1B_1^T)}{-Tr\,(A(F)K+KA(F)^T)}}, \tag{3.7}$$

which can be used to estimate γ for given F, L, and K. Since there is no guarantee for the fraction within the square root to be non-negative we need to take the absolute value to this fraction.

The algorithm corresponding to the case of γ being a parameter is stated in the following lines.

Algorithm 3.2. (FOM2: First-order method for solving the KKT system (2.7)-(2.8), and (2.10) - γ is a parameter)

Given $(F_0, L_0, \gamma_0) \in \mathcal{F}_s$ with $\gamma_0 > 0$, and $K_0 \succ 0$. Choose $\beta, \epsilon \in (0,1)$, $\tau \in (0, \frac{1}{2})$.

While the termination criterion (3.6) is not reached, do

a. Given L_k, K_k, then compute $F_{k+1} = \tilde{F}(L_k, K_k)$ by using (3.1).

b. Compute the spectral abscissa $\alpha(A(F_{k+1}))$.
   While $\alpha(A(F_{k+1})) \geq 0$, do
   (Perform backtracking)
   Set $l \leftarrow l+1$ and $F_{k+1} = F_k + \beta^l\tau(F_k - F_{k+1})$.
   If $\beta^l\tau \leq \epsilon$, stop.

c. Given F_{k+1}, solve the Lyapunov equation (3.2) for L_{k+1}.

d. Compute the new estimate $\gamma_{k+1} = \gamma(F_{k+1}, L_{k+1}, K_k)$ by using (3.7).

e. Solve the nonlinear matrix equation (3.4) in K_{k+1}.

f. Set $k \leftarrow k+1$ and go to step a.

End(While)

Next, regarding the initialization of the two methods FOM1 and FOM2 if A is such that $\alpha(A) < 0$, then we choose $F_0 = 0_{n_u \times n_y}$. Otherwise, we use any of the available software for computing suboptimal stabilizing SOF controller F_0 such that $\alpha(A(F_0)) < 0$, e.g., by using the Linear Matrix Inequality based method [24, slpmm].

We emphasize here that, currently a first-order penalty method is under investigation for computing suboptimal output feedback controllers. In that work we consider an unconstrained minimization problem of the form; see [34]:

$$\min_{(F,\mu)\in(\mathcal{S}_F,\mathbb{R}_+)} \quad J(F,\mu) = Tr(L(F,\mu)B_1B_1^T) + \sigma\mu^2, \tag{3.8}$$

where $L(F,\mu)$ solves the Lyapunov equation:

$$L\bar{A}(F,\mu) + \bar{A}(F,\mu)^TL + C(F)^TC(F) = 0, \tag{3.9}$$

where

$$\bar{A}(F,\mu) = A_\mu + BFC,$$
$$A_\mu = A \pm \mu I_{n_x}, \quad \mu \in \mathbb{R}_+,$$
$$C(F) = C_1 + D_1FC,$$

and (F,μ) belongs to the set

$$\mathcal{S}_F = \left\{(F,\mu) \in \mathbb{R}^{n_u \times n_y} \times \mathbb{R}_+ \mid \alpha(\bar{A}(F,\mu)) < 0\right\}, \tag{3.10}$$

σ is a large positive constant, and the dimensions of all matrices are as defined in Section 2. Clearly this problem is an unconstrained optimization problem in the unknowns F and μ.

Observe that in (3.8)-(3.10) we parameterize the constant matrix A and replace it by $A_\mu = A \pm \mu I_{n_x}$, where $\mu > 0$ is a scalar variable. In this approach we replace the matrix A by A_μ and include μ as a quadratic penalty term in the objective function. The sign in A_μ is chosen such that its spectral abscissa $\alpha(A_\mu)$ is strictly negative.

By applying the first-order necessary optimality conditions on this problem, the resulting nonlinear system can be solved using a similar approach to that used with the method FOM1. The advantage of parameterizing A is that we can initiate this method by

$$(\hat{F}_0, \gamma_0) = (0_{n_u \times n_y}, \gamma_0),$$

for some $\gamma_0 > 0$. Let the computed suboptimal stabilizing SOF gain be $\hat{F}$, where $\alpha(A(\hat{F})) < 0$. Consequently, this suboptimal SOF controller can be considered for initiating the two methods FOM1 and FOM2 as follows. Given $F_0 = \hat{F}$ with $\alpha(A(F_0))$ and for $\gamma_0 > 0$ one can obtain the initial L_0 and K_0 by solving modifications of the Lyapunov equations (2.10) and (2.11), respectively:

$$L\big(A(F_0) + \frac{1}{2\gamma_0^2}B_1B_1^T\big) + \big(A(F_0) + \frac{1}{2\gamma_0^2}B_1B_1^T\big)^T L + C(F_0)^T C(F_0) = 0, \tag{3.11}$$

$$A(F_0)K + KA(F_0)^T + \frac{1}{\gamma_0^2}\big(B_1B_1^T L_0 + L_0 B_1 B_1^T\big) = 0. \tag{3.12}$$

The appropriate values assigned to the parameters $\beta, \tau, \epsilon, \gamma_0$ of the algorithms FOM1 and FOM2 are given in Section 4.

4. Numerical Results

In this section, an implementation for the two algorithms FOM1 and FOM2 is described. All of the computations in this section were carried out on a PC with a Pentium IV 2.8 Ghz processor with 224 MB RAM. Two Matlab codes were written corresponding to this implementation. Within the two methods FOM1 and FOM2 we need to solve one nonlinear equation per iteration. It is desirable to compute approximate solution for that equation, which can be achieved by using the Matlab function fsolve from the Optimization Toolbox.

We have used the benchmark collection COMPl$_e$ib [25] for testing the two methods. In order to see the convergence behavior of the proposed methods some of the obtained results are listed in table forms, which are the following: the iterations counter k, the computed γ_k, the convergence criterion $\|\nabla_F \mathcal{L}_k\| + \|H_k\|$, the absolute value $|\Delta\gamma_k|$ of the change in γ_k, the number of iterations i_{nonlin} of the Matlab nonlinear solver fsolve required for solving the

Table 4.1: Performance of the method FOM1 on Example 4.1.

k	γ_k	$\|\nabla_F \mathcal{L}_k\| + \|H_k\|$	$\|\Delta\gamma_k\|$	i_{nonlin}	$\alpha(A(F_k))$
0	1.0000e+03	5.62e-04	-	0	-1.3e+00
1	1.5562e+00	1.57e-07	9.98e+02	4	-1.3e+00
2	1.0137e-02	1.08e-07	1.55e+00	4	-6.4e-01
3	9.5627e-03	1.03e-15	5.74e-04	12	-4.4e-01

Table 4.2: Performance of the method FOM1 on Example 4.2.

k	γ_k	$\|\nabla_F \mathcal{L}_k\| + \|H_k\|$	$\|\Delta\gamma_k\|$	i_{nonlin}	$\alpha(A(F_k))$
0	1.0000e+03	2.03e+01	-	0	-5.2e-01
1	1.8321e+01	3.77e-06	9.82e+02	28	-6.2e-02
2	1.4583e-02	2.26e-06	1.83e+01	98	-3.1e-01
3	8.1404e-03	1.45e-08	6.44e-03	21	-5.3e-01

nonlinear equation (3.4), and the spectral abscissa $\alpha(A(F_k))$ representing the stability margin of the closed-loop control system (2.3).

In the computations the following values have been assigned to the parameters of the two methods FOM1 and FOM2: $\gamma_0 = 10^3$, $\beta = 0.8$, $\tau = 0.4$, $\epsilon = 10^{-5}$, and $\epsilon_{tol} = 10^{-7}$.

4.1. Performance of the method FOM1

In order to see the convergence behavior of the method FOM1 we consider first in detail three test problems from the benchmark collection COMPl$_\text{e}$ib [25].

Example 4.1. The first example is a chemical reactor model [25, REA1]. The given data matrices are

$$A = \begin{bmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}, \quad C^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 0 & -1 \end{bmatrix},$$

$$B_1 = C_1 = D_1 = [\,],$$

where we borrow the Matlab notation [ ] to denote the empty matrix. As is described above, the main goal is to compute a feasible point (F_*, L_*, γ_*) to the NSDP problem (2.4) such that $\gamma_* > 0$ is attaining its least possible value. The zero matrix $F_0 = 0_{n_u \times n_y}$ is such that $\alpha(A(F_0)) \geq 0$. Therefore, the following F_0 is chosen:

$$F_0 = \begin{bmatrix} 0.357 & -2.6200 \\ 2.580 & 0.7764 \end{bmatrix}.$$

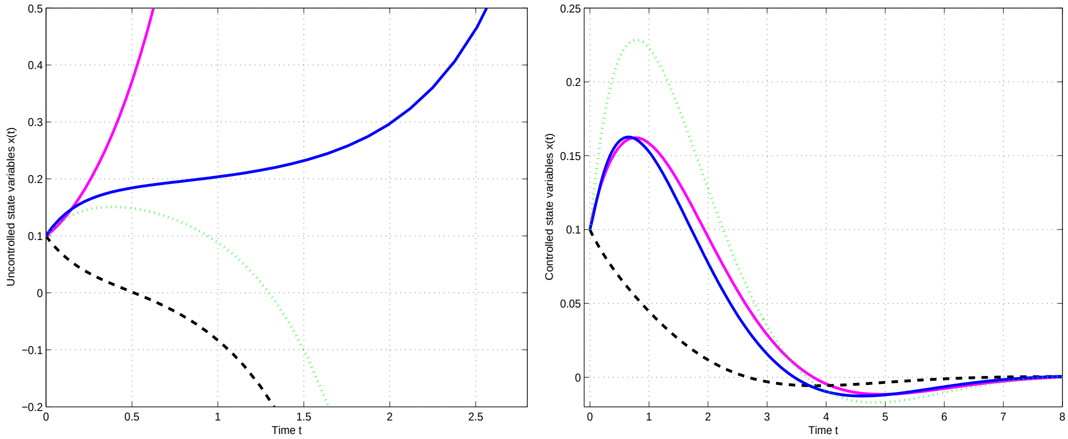

Fig. 4.1. Uncontrolled and controlled state space models for Example 4.1.

After 3 iterations and 20 iterations for the nonlinear solver (see Table 4.1) the method FOM1 converges to the stationary point (F_*, L_*, γ_*), where $\gamma_* = 9.5627 * 10^{-3}$,

$$F_* = \begin{bmatrix} -0.230 & 0.815 \\ 4.071 & -2.869 \end{bmatrix},$$

and L_* is the corresponding solution of the Lyapunov equation (3.11).

Fig. 4.1 shows the performance of the computed SOF controller F_* on the closed-loop control system (2.3), which enforces all state variables to the zero state.

Example 4.2. This test problem is a power system model [25, PSM]. The given data are the following

$$A = \begin{bmatrix} -0.042 & 0 & 4.92 & -4.92 & 0 & 0 & 0 \\ -5.21 & -12.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.33 & -3.33 & 0 & 0 & 0 & 0 \\ 0.545 & 0 & 0 & 0 & -0.545 & 0 & 0 \\ 0 & 0 & 0 & 4.92 & -0.042 & 0 & 4.92 \\ 0 & 0 & 0 & 0 & -5.21 & -12.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.33 & -3.33 \end{bmatrix},$$

$$B = \begin{bmatrix} -4.92 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -4.92 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad C^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 \\ 12.5 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 12.5 \\ 0 & 0 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} C & 0_{2\times 7} \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0_{3\times 2} \\ I_2 \end{bmatrix}.$$

In this example the zero matrix $F_0 = 0_{n_u \times n_y}$ is such that $(F_0, L_0, \gamma_0) \in \mathcal{F}_s$, where L_0 is the corresponding solution of the Lyapunov equation (3.11) and $\gamma_0 = 10^3$. The method FOM1

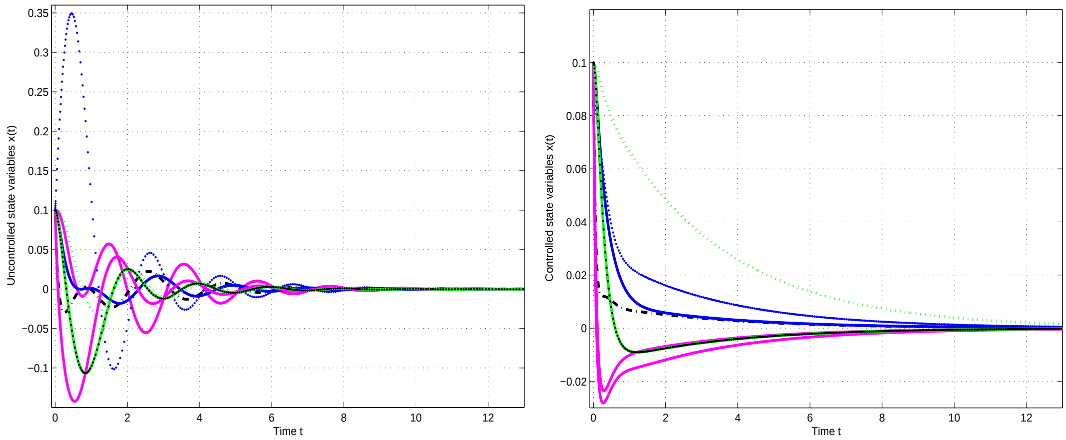

Fig. 4.2. Uncontrolled and controlled state space models for Example 4.2.

reaches the stationary point (F_*, L_*, γ_*) in 3 iterations and 147 inner iterations for the nonlinear solver (see Table 4.2) yielding the least value $\gamma_* = 8.1404 * 10^{-3}$ and the $\mathcal{H}_\infty$ stabilizing SOF controller

$$F_* = \begin{bmatrix} 3.249 & -0.840 & 2.242 \\ 0.255 & 0.086 & 2.533 \end{bmatrix}.$$

Table 4.2 shows the convergence behavior of the method FOM1 on this example. In Fig. 4.2 we see the effect of the computed SOF controller F_* on the closed-loop control system (2.3). Although the open-loop system is stable, the computed SOF controller F_* smears the oscillatory behavior of the state variables.

Example 4.3. This test example is an aircraft model [25, AC1]. It has the following data matrices

$$A = \begin{bmatrix} 0 & 0 & 1.132 & 0 & -1 \\ 0 & -0.054 & -0.171 & 0 & 0.071 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.049 & 0 & -0.856 & -1.013 \\ 0 & -0.291 & 0 & 1.053 & -0.686 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ -0.12 & 1 & 0 \\ 0 & 0 & 0 \\ 4.419 & 0 & -1.665 \\ 1.575 & 0 & -0.073 \end{bmatrix},$$

$$C^T = \begin{bmatrix} I_3 \\ 0_{2\times 3} \end{bmatrix}, \quad B_1 = \begin{bmatrix} 3.59e-2 & 0 & 1.672e-2 \\ 0 & 9.89e-3 & 0 \\ 0 & -7.55e-2 & 0 \\ 0 & 0 & 5.635e-2 \\ 1.45e-3 & 0 & 6.743e-2 \end{bmatrix},$$

$$C_1^T = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The initial iterate $(F_0, L_0, \gamma_0) \in \mathcal{F}_s$ is chosen, where $\gamma_0 = 10^3$,

$$F_0 = \begin{bmatrix} 0.8221 & -0.0173 & 0.3160 \\ 0.0183 & -1.7067 & 0.0438 \\ 2.1731 & 0.0233 & 1.8150 \end{bmatrix},$$

Table 4.3: Performance of the method FOM1 on Example 4.3.

k	γ_k	$\|\nabla_F \mathcal{L}_k\| + \|H_k\|$	$\|\Delta\gamma_k\|$	i_{nonlin}	$\alpha(A(F_k))$
0	1.0000e+03	1.80e+01	-	0	-2.2e-01
1	1.4767e+01	3.27e-06	9.85e+02	41	-2.4e-01
2	2.8094e-02	8.58e-07	1.47e+01	18	-1.5e-01
3	1.0457e-02	4.45e-10	1.76e-02	4	-1.4e-01

Table 4.4: Performance of the method FOM2 on Example 4.4.

k	γ_k	$\|\nabla_F \mathcal{L}_k\| + \|H_k\|$	$\|\Delta\gamma_k\|$	i_{nonlin}	$\alpha(A(F_k))$
0	1.0000e+03	1.74e+03	-	0	-3.9e-01
1	4.6770e-02	1.45e-06	1.00e+03	114	-1.3e+00
2	2.7607e-02	5.12e-08	1.92e-02	18	-2.2e-01

and L_0 is the corresponding solution of the Lyapunov equation (3.11). After 3 iterations and 63 inner iterations of the nonlinear solver fsolve the method FOM1 reaches the least $\gamma_* = 1.0457 * 10^{-2}$ and the corresponding $\mathcal{H}_\infty$ stabilizing SOF feedback gain:

$$F_* = \begin{bmatrix} 0.226 & -0.253 & -0.149 \\ 0.238 & -1.060 & 0.103 \\ 1.306 & -1.321 & 0.676 \end{bmatrix}.$$

Table 4.3 shows the convergence behavior of the method FOM1 on this example. Fig. 4.3 shows the uncontrolled and controlled state variables.

In addition to the above results, Table 4.6 gives some preliminary tests for the method FOM1 in a compact form. For every test problem we report the problem name together with the problem dimensions (n_x, n_u, n_y, n_w, n_z), the total CPU time in seconds, the least computed γ_*, and the overall number of iterations together with the number of inner iterations by the nonlinear solver fsolve.

Note, however, that for some test problems the matrix D_1 does not exist or sometimes $D_1^T D_1$ is not invertible (see [25]). In such cases we replace $D_1^T D_1$ by the identity matrix or any positive definite matrix.

For the sake of comparison, we have compared the computed γ_* using the method FOM1 of Table 4.6 with its corresponding counterpart that was obtained in [22, Table 2], where an augmented Lagrangian method was used with the publicly available general purpose software PENBMI. The method FOM1 wins out of 55 test problems in 37 trials to reach a lower value for γ_* and consequently a better stationary point (F_*, L_*, γ_*), while the augmented Lagrangian PENBMI method of [22] wins in only 18 trials.

4.2. Performance of the method FOM2

In this subsection we give some preliminary tests of the method FOM2 for solving the NSDP problem (2.4). The convergence behavior of the method FOM2 can be seen through the following examples.

Example 4.4. The considered test problem is a decentralized interconnected system [25, DIS3].

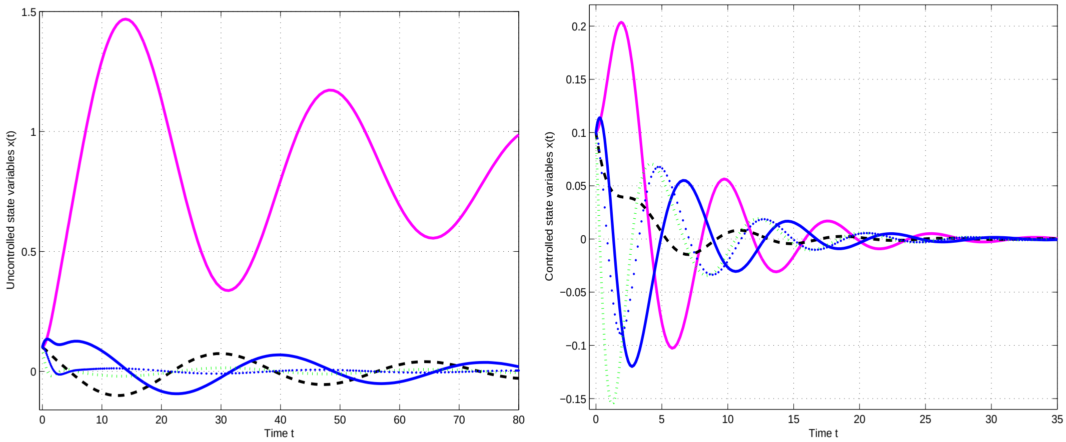

Fig. 4.3. Uncontrolled and controlled state space models for Example 4.3.

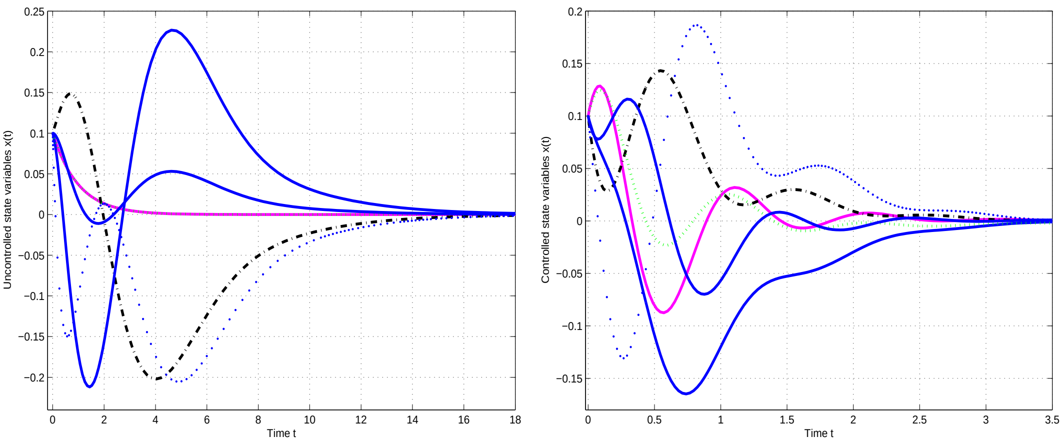

Fig. 4.4. Uncontrolled and controlled state space models for Example 4.4.

The linearized state space model has the following given data

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -2 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ -8 & 1 & -1 & -1 & -2 & 0 \\ 4 & -0.5 & 0.5 & 0 & 0 & -4 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C^T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In this test problem (see [25]), $B_1 = C_1 = D_1 = [\,]$.

Table 4.5: Performance of FOM2 on test problem [25, AC15]: updating γ using (4.1).

k	γ_k	$\|\nabla_F \mathcal{L}_k\| + \|H_k\|$	$\|\Delta\gamma_k\|$	i_{nonlin}	$\alpha(A(F_k))$
0	1.0000e+03	5.20e-04	–	0	-3.6e-01
1	7.0592e+02	2.65e-18	2.94e+02	2	-3.4e-01
2	4.9911e+02	1.63e-18	2.07e+02	2	-3.5e-01
3	3.5322e+02	1.21e-18	1.46e+02	2	-3.5e-01
⋮	⋮	⋮	⋮	⋮	⋮
13	1.7803e+01	3.71e-18	1.96e+00	3	-4.0e-01
14	1.7719e+01	1.43e-17	8.45e-02	4	-5.0e-01

The following starting point is chosen $(F_0, L_0, \gamma_0) = (0_{n_u \times n_y}, L_0, 10^3) \in \mathcal{F}_s$, where L_0 is the corresponding solution of the Lyapunov equation (3.11). Starting from that point, the method FOM2 converges after two iterations and 132 inner iterations of the nonlinear solver fsolve to the stationary point (F_*, L_*, γ_*), where $\gamma_* = 2.7607 * 10^{-2}$,

$$F_* = \begin{bmatrix} -199.274 & -104.219 & -9.270 & 59.588 \\ -132.158 & -149.031 & -40.432 & 56.895 \\ 51.300 & -42.678 & -42.357 & 17.619 \\ 117.298 & 15.543 & -28.145 & -31.296 \end{bmatrix},$$

and L_* is the corresponding solution of the Lyapunov equation (3.11).

Table 4.4 shows the convergence behavior of the method FOM2 on this example. Fig. 4.4 shows the uncontrolled and controlled state variables.

Additional information regarding the performance of the methods FOM1 and FOM2 for all the above test problems can be found within Tables 4.6 and 4.7. If we compare the computed γ_* of Table 4.7 with the corresponding values obtained in [22, Table 2] we find that the method FOM2 wins out of 55 test problems in 28 trials to reach better γ_*'s, while the augmented Lagrangian algorithm with PENBMI [22] wins in 27 trials. In Table 4.7, the test problem DIS3 was tested using two different starting points; the associated results correspond to the second starting point.

We have also tried to estimate γ within the method FOM2 using

$$\gamma(F, L, K) = \sqrt{\frac{|Tr\,(LB_1B_1^T L)|}{|-Tr\,(LA(F) + A(F)^T L + C(F)C(F)^T)|}}, \tag{4.1}$$

which is obtained from (2.10) similar to (3.7). The absolute value is taken to the fraction within the square root to ensure it to be non-negative. The method FOM2 when using (4.1) is less efficient. A typical convergence behavior in this case on most test problems is shown in Table 4.5.

Observe that the considered approach is first-order and iterative, where a nonlinear matrix equation is solved in each iteration. Tables 4.1-4.3 and 4.4 show the convergence behavior of the methods FOM1 and FOM2 for computing a stationary point of the NSDP problem (2.4), respectively.

Tables 4.6 and 4.7 show the efficiency and robustness of the proposed methods, where the number of iterations in the last column of these tables show the efficiency of these methods. In Table 4.6 exactly 51 test problems out of 55 did not exceed 5 outer iterations to reach the

Table 4.6: Performance of the method FOM1 on test problems from [25].

Problem	n_x	n_u	n_y	n_w	n_z	CPU(sec)	γ_*	$\alpha(A(F_k))$	No. of iter.
AC1	5	3	3	3	2	4.51e-01	5.1438e-02	-4.4e-02	3(47)
AC2	5	3	3	5	5	6.11e-01	3.1569e-02	-3.3e-01	3(71)
AC3	5	2	4	5	5	1.50e+00	1.6228e-01	-2.9e-01	3(101)
AC4	4	1	2	2	2	3.40e-01	2.8771e-02	-5.0e-02	3(95)
AC6	7	2	4	7	7	7.01e-01	3.5232e-02	-4.0e-01	3(55)
AC7	9	1	2	4	1	8.73e+00	6.0423e-01	-4.1e-02	3(306)
AC8	9	1	5	10	2	7.91e-01	8.6637e-03	-2.2e-01	3(29)
AC9	10	1	5	1	10	2.11e+01	6.5976e-01	-1.1e-01	4(473)
AC11	4	2	4	4	4	2.81e+00	2.5841e-01	-8.3e-02	3(264)
AC12	4	3	4	3	1	5.91e-01	1.8739e+01	-6.1e-02	2(51)
AC15	4	2	3	4	6	3.00e-01	3.4417e-03	-1.0e-01	2(20)
AC16	4	2	4	4	6	2.01e-01	5.0301e-02	-3.3e-01	2(9)
AC17	4	1	2	4	4	4.51e-01	3.3488e-02	-4.7e-01	2(49)
HE1	4	2	1	2	2	1.80e-01	3.6161e-02	-8.5e-02	2(12)
HE2	4	2	2	4	4	2.90e+00	2.9817e-02	-1.5e-02	4(374)
HE3	8	4	6	1	10	2.08e+00	1.0912e+01	-1.7e-03	7(150)
HE4	8	4	6	8	12	3.82e+00	2.7568e+00	-2.9e-02	4(187)
HE5	8	4	2	3	4	2.28e+01	1.4431e+02	-1.1e-05	9(902)
REA1	4	2	3	4	4	2.30e-01	9.5627e-03	-1.1e+00	3(30)
REA2	4	2	2	4	4	7.01e-01	2.8419e-02	-4.2e-01	3(69)
REA3	12	1	3	12	12	2.34e+01	3.4095e+00	-2.1e-02	5(370)
DIS1	8	4	4	1	8	3.51e-01	3.5917e-02	-1.4e-01	3(9)
DIS2	3	2	2	3	3	2.61e-01	6.5138e-02	-1.4e+00	5(24)
DIS3	6	4	4	6	6	5.71e-01	6.2902e-03	-6.4e-01	5(52)
DIS4	6	4	3	6	6	2.76e+00	2.1223e-01	-3.0e-02	3(197)
DIS5	4	2	2	3	3	1.55e+00	3.6074e+03	-9.4e-01	2(221)
WEC1	10	3	4	10	10	1.21e+01	5.7023e-02	-6.7e-02	5(337)
WEC2	10	3	4	10	10	1.39e+01	1.9540e-01	-7.8e-01	5(360)
WEC3	10	3	4	10	10	1.05e+01	2.2362e-01	-1.3e+00	3(325)
UWV	8	2	2	2	1	5.80e+00	1.5341e-02	-6.3e-02	3(245)
CSE1	20	2	10	1	12	9.66e+00	1.7930e-01	-8.7e-02	3(21)
EB1	10	1	1	2	2	1.27e+01	8.8455e-01	-1.5e-01	3(286)
EB2	10	1	1	2	2	7.61e-01	1.1365e-02	-1.1e-03	3(15)
TF1	7	2	4	1	4	3.57e+00	9.9789e-02	-2.3e-02	3(154)
TF2	7	2	3	1	4	2.68e+00	9.5752e-01	-1.0e-05	5(147)
TF3	7	2	3	1	4	4.62e+00	2.0318e-01	-3.2e-03	5(246)
PSM	7	2	3	2	3	2.68e+00	8.1404e-03	-5.3e-01	3(147)
BDT1	11	3	3	1	6	9.61e-01	7.6094e-01	-5.6e-04	3(12)
MFP	4	2	3	4	4	3.11e-01	1.9536e-01	-4.2e-03	3(22)
TG1	10	2	2	10	10	1.19e+01	1.9325e-02	-1.9e-01	5(294)
AGS	12	2	2	12	12	4.54e+00	1.9708e-02	-1.7e-01	3(80)
NN1	3	1	2	3	3	5.61e-01	2.7681e-02	-1.2e+00	3(15)
NN2	2	2	2	2	2	7.00e-02	4.4387e-03	-2.2e-01	3(15)
NN4	4	2	3	4	4	2.31e-01	5.7396e-03	-1.9e-01	3(19)
NN5	7	1	2	7	7	2.31e+00	3.7458e-01	-7.0e-04	5(146)
NN6	9	1	4	9	9	1.27e+01	1.2868e+01	-1.1e-02	5(394)
NN7	9	1	4	5	3	1.63e+00	8.5178e-01	-1.3e-02	3(82)
NN9	5	3	2	2	4	1.63e+00	3.1553e+00	-1.2e-01	3(245)
NN11	16	3	5	3	3	1.01e+01	2.6512e+01	-8.3e-01	13(26)
NN12	6	2	2	6	6	1.92e+00	3.8781e-01	-2.1e-01	3(149)
NN13	6	2	2	3	3	2.75e+00	2.7177e-02	-5.8e-01	3(227)
NN14	6	2	2	3	3	2.03e+00	2.0957e-02	-4.7e-01	3(175)
NN15	3	2	2	1	4	7.51e-01	4.4143e+01	-1.2e-01	5(168)
NN16	8	4	4	8	4	1.27e+00	1.0179e-01	-4.4e-02	3(78)
NN17	3	2	1	1	2	2.10e-01	4.0159e-02	-4.5e-01	3(21)

corresponding stationary point (F_*, L_*, γ_*), while in Table 4.7 the number of iterations did not exceed 4 outer iterations in the whole 55 test problems. On the other hand, the total CPU time for all test examples (the 7th column of both of Tables 4.6 and 4.7) is less than 45 seconds in the worst case of Table 4.6 and 11 seconds in Table 4.7, which also show the robustness of the proposed methods for determining stationary points of the considered problems.

Table 4.7: Performance of the method FOM2 on test problems from [25].

Problem	n_x	n_u	n_y	n_w	n_z	CPU(sec)	γ_*	$\alpha(A(F_k))$	No. of iter.
AC1	5	3	3	3	2	9.31e-01	1.6688e+00	-2.0e-01	2(104)
AC2	5	3	3	5	5	5.71e-01	4.1406e+00	-2.0e-01	3(55)
AC3	5	2	4	5	5	3.30e-01	1.2278e+00	-4.2e-01	2(45)
AC4	4	1	2	2	2	9.51e-01	2.3987e+00	-5.0e-02	3(153)
AC6	7	2	4	7	7	2.30e+00	7.5352e-01	-1.8e-01	2(123)
AC7	9	1	2	4	1	3.61e+00	2.8690e+01	-3.9e-02	2(122)
AC8	9	1	5	10	2	8.71e-01	5.6789e-01	-2.1e-01	2(36)
AC9	10	1	5	1	10	1.07e+01	1.4853e+01	-6.4e-02	3(253)
AC11	4	2	4	4	4	2.41e-01	1.1965e+00	-1.2e-01	1(23)
AC12	4	3	4	3	1	1.50e-01	2.6642e+01	-4.4e-02	2(24)
AC15	4	2	3	4	6	7.00e-02	1.5782e+01	-2.2e-01	2(6)
AC16	4	2	4	4	6	9.00e-02	1.0039e+01	-3.2e-01	2(7)
AC17	4	1	2	4	4	1.70e-01	8.6284e-01	-4.9e-01	2(20)
HE1	4	2	1	2	2	1.81e-01	1.7026e+00	-4.9e-02	2(18)
HE2	4	2	2	4	4	1.41e-01	1.6350e+00	-1.8e-01	1(15)
HE3	8	4	6	1	10	3.10e-01	1.5968e+01	-7.1e-03	1(24)
HE4	8	4	6	8	12	6.51e-01	6.0622e+00	-1.1e-02	1(34)
HE5	8	4	2	3	4	7.90e+00	5.8318e-01	-1.1e-01	3(335)
REA1	4	2	3	4	4	8.00e-02	5.8107e-01	-3.6e+00	1(14)
REA2	4	2	2	4	4	2.50e-01	1.7508e-01	-1.9e+00	2(37)
REA3	12	1	3	12	12	8.12e+00	5.5686e+00	-2.1e-02	1(116)
DIS1	8	4	4	1	8	2.40e-01	2.3621e+00	-5.1e-01	1(9)
DIS2	3	2	2	3	3	4.00e-01	1.4505e+00	-3.6e+00	1(12)
DIS3	6	4	4	6	6	8.00e-01	1.1746e+00	-1.3e+00	1(13)
DIS3[1)]	6	4	4	6	6	1.81e+00	2.7607e-02	-2.2e-01	2(132)
DIS4	6	4	3	6	6	2.49e+00	4.8810e+00	-1.6e-01	2(179)
DIS5	4	2	2	3	3	5.81e-01	1.7725e+01	-9.6e-01	1(73)
WEC1	10	3	4	10	10	5.64e+00	2.5431e-01	-2.1e-01	2(160)
WEC2	10	3	4	10	10	7.31e-01	3.7787e-01	-4.4e-01	1(20)
WEC3	10	3	4	10	10	1.01e+01	2.7003e-01	-7.9e-01	2(228)
UWV	8	2	2	2	1	3.28e+00	2.1567e-01	-1.4e+01	2(146)
CSE1	20	2	10	1	12	3.09e+00	7.1648e+00	-5.5e-02	1(6)
EB1	10	1	1	2	2	6.11e-01	1.5620e+00	-1.5e-01	1(13)
EB2	10	1	1	2	2	4.91e-01	3.0089e+00	-1.0e-01	1(11)
TF1	7	2	4	1	4	1.39e+00	7.6462e+00	-2.4e-02	4(63)
TF2	7	2	3	1	4	-	-	-	-
TF3	7	2	3	1	4	2.69e+00	1.2995e+02	-3.2e-03	2(146)
PSM	7	2	3	2	3	9.41e-01	3.1524e-01	-6.2e-02	1(54)
BDT1	11	3	3	1	6	6.31e-01	1.6874e+02	-1.1e-03	2(10)
MFP	4	2	3	4	4	2.11e-01	2.2419e+01	-1.2e-02	2(12)
TG1	10	2	2	10	10	7.71e+00	4.7523e+00	-4.8e-01	2(180)
AGS	12	2	2	12	12	3.38e+00	1.8872e-01	-1.9e-01	3(71)
NN1	3	1	2	3	3	1.71e-01	1.9161e+00	-3.3e-01	2(15)
NN2	2	2	2	2	2	4.00e-02	1.4564e+00	-3.8e-01	1(4)
NN4	4	2	3	4	4	1.30e-01	1.2858e+00	-3.9e-01	1(8)
NN5	7	1	2	7	7	8.21e-01	5.5475e+01	-3.3e-02	1(44)
NN6	9	1	4	9	9	6.62e+00	5.5569e+00	-4.2e-02	2(219)
NN7	9	1	4	5	3	1.53e+00	3.4646e+00	-3.2e-02	2(67)
NN9	5	3	2	2	4	3.25e+00	5.5859e+00	-5.1e-02	3(341)
NN11	16	3	5	3	3	3.10e-01	2.6633e+00	-8.7e-01	1(2)
NN12	6	2	2	6	6	1.47e+00	1.2654e+01	-3.4e-02	1(104)
NN13	6	2	2	3	3	3.61e-01	8.2769e-01	-1.5e+00	1(22)
NN14	6	2	2	3	3	1.52e+00	8.2088e-01	-1.5e+00	1(104)
NN15	3	2	2	1	4	2.81e-01	3.0855e+01	-1.5e+00	1(33)
NN16	8	4	4	8	4	1.29e+00	4.7154e+00	-3.0e-02	2(57)
NN17	3	2	1	1	2	1.50e-01	5.4893e+00	-4.9e-01	3(14)

By comparing the obtained results of Tables 4.6 and 4.7, we observe the following: $\gamma_*^{FOM1} < \gamma_*^{FOM2}$ for most of the considered test problems indicating that FOM1 achieves better stationary points (F_*, L_*, γ_*) than FOM2. On the other side, the number of iterations is remarkably reduced when employing the method FOM2 on most of the considered test problems. Therefore, one cannot give a specific decision which one of the two algorithms outperforms the other.

References

[1] P. Apkarian and P. Gahinet, A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers, *IEEE T. Automat. Control*, **40** (1995), 853-864.

[2] P. Apkarian, D. Noll and H.D. Tuan, Fixed-order $\mathcal{H}_\infty$ control design via augmented Lagrangian method, *Int. J. Robust Nonlinear Control*, **13** (2003), 1137-1148.

[3] P. Apkarian, D. Noll, J.-B. Thevenet and H.D. Tuan, A spectral quadratic-SDP method with applications to fixed-order $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis, in Proceedings Asian Control Conf., Melbourne, Australia, 2004.

[4] D.S. Bernstein and W.M. Haddad, LQG control with an $\mathcal{H}_\infty$ performance bound: A Riccati equation approach, *IEEE T. Automat. Control*, **34** (1989), 293-305.

[5] S.P. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory, Vol. **15** of SIAM Studies in Applied Mathematics, SIAM, Philadelphia, 1994.

[6] B.M. Chen, $\mathcal{H}_\infty$ control and its applications, Vol. 235 of LNCIS, Springer-Verlag, Berlin, New York, 1998.

[7] M. Chilali and P. Gahinet, $\mathcal{H}_\infty$ design with pole placement constraints: An LMI approach, *IEEE T. Automat. Control*, **41** (1996), 358-367.

[8] J. Doyle, K. Glover, P.P. Khargoneker, and B.A. Francis, State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems, in Proc. American Control Conf., 1988, 1691-1696.

[9] L. El Ghaoui, F. Oustry and M. Ait Rami, 1997. A cone complementarity linearization algorithm for static output-feedback and related problem, *IEEE T. Automat. Control*, **42** (1997), 1171-1176.

[10] B. Fares, P. Apkarian and D. Noll, An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory, *Int. J. Control*, **74** (2001), 348-360.

[11] B. Fares, D. Noll and P. Apkarian, Robust control via sequential semidefinite programming, *SIAM J. Control Optimiz.*, **40** (2002), 1791-1820.

[12] P. Gahinet and P. Apkarian, A linear matrix inequality approach to $\mathcal{H}_\infty$ control, *Int. J. Robust Nonlinear Control*, **4** (1994), 421-448.

[13] K.C. Goh, M.G. Safonov and J.H. Ly, Robust synthesis via bilinear matrix inequality, *Int. J. Robust Nonlinear Control*, **6** (1996), 1079-1095.

[14] K.M. Grigoriadis and E.B. Beran, Alternating projection algorithms for linear matrix inequalities problems with rank constraints, in Advances on Linear Matrix Inequality Methods in Control, L. El Ghaoui and S.-I. Niculescu (Eds.), SIAM, 1999, 251-267.

[15] K.M. Grigoriadis and R. Skelton, Low-order control design for LMI problems using alternating projection methods, *Automatica*, **32** (1996), 1117-1125.

[16] H. Hassibi, J. How and S. Boyd, A path following method for solving BMI problems in control, in Proc. American Control Conf., 1999, 1385-1389.

[17] C.W.J. Hol and C.W. Scherer, A sum of squares approach to fixed order $\mathcal{H}_\infty$ synthesis, Research Rep., Delft Univ., in D. Henrion and A. Garulli (Eds.). Positive Polynomials in Control, LNCIS 312, Springer-Verlag, Berlin, New York, 2005, 45-71.

[18] S. Ibaraki and M. Tomizuka, Rank minimization approach for solving BMI problems with random search, in Proc. American Control Conf., 2001, 25-27.

[19] T. Iwasaki and R.E. Skelton, Parameterization of all stabilizing controllers via quadratic Lyapunov functions, *J. Optimiz. Theory Appl.*, **85** (1995), 291-307.

[20] S. Kanev, C. Sherer, M. Verhaegen and B.De Schutter, Robust output feedback controller design via a local BMI optimization, *Automatica*, **40** (2004), 1115-1127.

[21] P.P. Khargonekar and M.A. Rotea, Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: A convex optimization approach, *IEEE T. Automat. Control*, **36** (1991), 824-837.

[22] M. Kočvara, F. Leibfritz, M. Stingl and D. Henrion, A nonlinear SDP algorithm for static output feedback problems in COMPl$_e$ib, LAAS-CNRS Research Report No. 04508, LAAS Toulouse, October 2004.

[23] W.S. Levine and M. Athans, On the determination of the optimal constant output feedback gains for linear multivariable systems, *IEEE T. Automat. Control*, **15** (1970), 44-48.

[24] F. Leibfritz, A LMI-based algorithm for designing suboptimal static $\mathcal{H}_2/\mathcal{H}_\infty$ output feedback xcontrollers, *SIAM J. Control Optimiz.*, **39** (2001), 1711-1735.

[25] F. Leibfritz, COMPl$_\mathrm{e}$ib: Constraint Matrix–Optimization Problem library – a collection of test examples for nonlinear semi-definite programs, control system design and related problems. Tech. Report-03, 2003.

[26] F. Leibfritz and E.M.E. Mostafa, Trust region methods for solving the optimal output feedback design problem, *Int. J. Control*, **76** (2003), 501-519.

[27] F. Leibfritz and E.M.E. Mostafa, An interior-point trust region method for a special class of nonlinear SDP-problems, *SIAM J. Optimiz.*, **12** (2002), 1048-1074.

[28] E.M.E. Mostafa, Efficient trust region methods in numerical optimization, Ph.D. thesis, Dept. of Mathematics, Faculty of Science, Alexandria University, Egypt, 2000.

[29] E.M.E. Mostafa, A trust region method for solving the decentralized static output feedback design problem, *J. Appl. Math. Comput.*, **18** (2005), 1-23.

[30] E.M.E. Mostafa, An augmented Lagrangian SQP method for solving some special class of nonlinear semidefinite programming problems, *Comput. Appl. Math.*, **24**(3) (2005), 461-486.

[31] D. Noll, M. Torki and P. Apkarian, Partially augmented Lagrangian method for matrix inequality constraints, *SIAM J. Optimiz.*, **15** (2004), 161-184.

[32] R. Orsi, U. Helmke and J.B. Moore, A Newton-like method for solving rank constrained linear matrix inequalities, in Proc. 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas, (2004), 3138-3144.

[33] E.D. Sonntag, Mathematical Control Theory, Springer-Verlag, New York, Berlin, 1990.

[34] V.L. Syrmos C.T. Abdallah, P. Dorato and K. Grigoriadis, Static output feedback: A survey, *Automatica*, **33** (1997), 125-137.