Journal of the Operations Research
Society of Japan
2008, Vol. 51, No. 1, 15-28

COMPUTATIONAL DESIGN OF OPTIMAL DISCRETE-TIME OUTPUT FEEDBACK CONTROLLERS

El–Sayed M.E. Mostafa
Alexandria University

(Received February 6, 2006; Revised June 17, 2007)

Abstract This paper considers the problem of designing a stabilizing static output feedback controller of linear discrete-time systems that minimizes certain quadratic performance index. A trust-region method is developed to solve an equivalent optimization problem of this optimal control problem. In addition, a first-order method is introduced to compute suboptimal stabilizing output feedback controllers that are used to initiate the trust-region method. Finally some numerical results that illustrate the performance of the proposed methods are given.

Keywords: Nonlinear programming, discrete-time linear quadratic control, trust-region methods, line search globalization

1. Introduction

The static output feedback control problem has received considerable attention in the control literature; see for instance the survey papers [12] and [19] and the references therein. In this paper we discuss the numerical solution of the optimal linear-quadratic discrete-time static output feedback problem of the form:

$$\begin{aligned} &\text{minimize} \quad J := \mathbb{E}\Big\{ \sum_{k=0}^{\infty} [x_k^T Q x_k + u_k^T R u_k] \Big\}, \\ &\text{subject to} \quad x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k, \end{aligned} \tag{1.1}$$

where $\mathbb{E}\{\cdot\}$ is the expected value; $x_k \in \mathbb{R}^{n_x}, u_k \in \mathbb{R}^{n_u}, y_k \in \mathbb{R}^{n_y}$ are the state, the control input, and the measured output vectors, respectively; $A \in \mathbb{R}^{n_x \times n_x}, B \in \mathbb{R}^{n_x \times n_u}, C \in \mathbb{R}^{n_y \times n_x}$ are known constant matrices, and $Q \in \mathbb{R}^{n_x \times n_x}, R \in \mathbb{R}^{n_u \times n_u}$ are given symmetric and positive semi-definite, positive definite weight matrices, respectively.

We consider the control law:

$$u_k = F y_k, \tag{1.2}$$

where $F \in \mathbb{R}^{n_u \times n_y}$ is the feedback gain matrix. Our main goal is to compute an optimal F that minimizes the above quadratic performance index and at the same time F must stabilize the associated closed-loop discrete-time system $x_{k+1} = (A + BFC)x_k$. Obviously, this recursive relation is equivalently written as $x_k = (A + BFC)^k x_0$, which restricts F to lie within the following set:

$$\mathcal{S}_F = \Big\{ F \in \mathbb{R}^{n_u \times n_y} \ : \ \rho\,(A + BFC) < 1 \Big\} \tag{1.3}$$

so that all state variables to be steered to the zero state, where $\rho\,(\cdot)$ is the spectral radius.

In that case $\lim_{k\to\infty} x_k = 0$. Note that the initial condition $x_0 \in \mathbb{R}^{n_x}$ is assumed to be a random variable satisfying $\mathbb{E}\{x_0\} = 0$.

The optimal control problem (1.1)–(1.2) can be formulated as an unconstrained minimization problem of the form (see, e.g., the two surveys [12], [19]):

$$\min_{F\in\mathcal{S}_F} \quad J(F) := \mathrm{Tr}\,(P(F)Q(F)), \tag{1.4}$$

where $P(F)$ solves the following discrete Lyapunov equation:

$$P(F) = A(F)P(F)A(F)^T + V, \tag{1.5}$$

$Q(F) := Q+C^TF^TRFC$, $A(F) := A+BFC$ and $V := \mathbb{E}\{x_0x_0^T\}$ is a given covariance matrix. The constant covariance matrix V is often chosen to be the identity or any symmetric positive definite matrix. The problem (1.4)–(1.5) can be considered as an unconstrained optimization problem in the matrix variable F, where the eigenvalue condition $F \in \mathcal{S}_F$ will not be treated as an explicit constraint rather will be fulfilled within the proposed method.

Trust-region methods have shown to be quite successful in solving various formulations of continuous-time (static/reduced order) output feedback control problems (see [13], [7], [8], [14], [15]). However, trust-region methods have not yet been employed to solve optimization problems arising from discrete-time control applications. In this paper we discuss the numerical solution of the optimization problem (1.4)–(1.5) by using a trust-region method. Interested reader is referred for instance to the book of Conn, Gould and Toint [2] for a survey of trust-region methods.

The existence of suboptimal stabilizing output feedback controllers is of great practical importance. Specially it can be used to initiate those methods that seek optimal solutions. Recently several methods have been investigated to determine suboptimal stabilizing controllers; see, e.g., [3], [11], [18], [6], [10]. A first-order method is developed to compute a suboptimal discrete stabilizing output feedback controller.

This paper is organized as follows. In the next section we state some basic results on the problem (1.4)–(1.5) required in constructing our method. In Section 3 we propose the trust-region algorithm (denoted by disTR) for solving (1.4)–(1.5). In Section 4 we state a first-order method for determining a suboptimal discrete stabilizing output feedback controller that can be used to initiate the trust-region method disTR. In Section 5 we test numerically the performance of the method disTR through several test problems from the literature. In addition, we compare our results with Newton's method combined with Armijo globalization strategy stated in [12].

Notations: Throughout the paper $\|\cdot\|$ denotes the Frobenius norm given by $\|M\| = \sqrt{\langle M, M\rangle}$, where $\langle\cdot,\cdot\rangle$ is the inner product defined by $\langle M_1, M_2\rangle = \mathrm{Tr}\,(M_1^TM_2)$ for $M_1, M_2 \in \mathbb{R}^{n\times n}$ and $\mathrm{Tr}\,(\cdot)$ is the trace operator. I_n denotes the $n \times n$ identity matrix.

2. Basic Results

This section is devoted to the evaluation of the first- and second-order directional derivatives of the objective function $J(F)$. The next two lemmas contain these derivatives. Similar results can be found in [12] and therefore we omit the proofs.

Lemma 2.1 *Let $F \in \mathcal{S}_F$. The first-order directional derivative of $J(F)$ in the direction of ΔF is given by*

$$J_F(F)\Delta F = 2\,\mathrm{Tr}\,((B^TS(F)A(F) + RFC)P(F)C^T\Delta F^T), \tag{2.1}$$

where $P(F)$ and $S(F)$ solve, respectively, the discrete Lyapunov equations (1.5) *and*

$$S(F) = A(F)^T S(F) A(F) + Q(F). \tag{2.2}$$

From the definition of the inner product and (2.1) we can express the gradient of $J(F)$ explicitly as:

$$\nabla J(F) = 2\,(B^T S(F) A(F) + RFC) P(F) C^T. \tag{2.3}$$

Lemma 2.2 *Let $F \in \mathcal{S}_F$, and let $P(F)$ and $S(F)$ be solutions to the discrete Lyapunov equations* (1.5) *and* (2.2)*, respectively. The second-order directional derivatives of $J(F)$ is given by*

$$\begin{aligned} J_{FF}(F)(\Delta F, \Delta F) \;&=\; \mathrm{Tr}\,(\Delta F^T (B^T S(F) B + R) \Delta F C P(F) C^T) \\ &\quad + 2\,\mathrm{Tr}\,(\Delta F^T B^T \Delta S(\Delta F) A(F) P(F) C^T), \end{aligned} \tag{2.4}$$

where $\Delta S(\Delta F)$ solves the discrete Lyapunov equation

$$\begin{aligned} \Delta S(\Delta F) \;&=\; A(F)^T \Delta S(\Delta F) A(F) + C^T \Delta F^T (B^T S(F) A(F) + RFC) \\ &\quad + (B^T S(F) A(F) + RFC)^T \Delta F C. \end{aligned} \tag{2.5}$$

Now we can form the second-order Taylor expansion of $J(F + \Delta F)$ around F:

$$J(F + \Delta F) = J(F) + J_F(F)\Delta F + \frac{1}{2} J_{FF}(F)(\Delta F, \Delta F) + o(\|\Delta F\|^2),$$

where

$$J_F(F)\Delta F = \mathrm{Tr}\,(\Delta F^T \nabla J(F)),$$

$o\,(\|\Delta F\|^2)$ is such that $\lim_{\|\Delta F\| \to 0} o\,(\|\Delta F\|^2)/\|\Delta F\|^2 = 0$, and $J_{FF}(F)(\Delta F, \Delta F)$ is given by (2.4). Therefore, the quadratic model $q(\Delta F)$ of $J(F + \Delta F)$ is

$$q(\Delta F) = J_F(F)\Delta F + \frac{1}{2} J_{FF}(F)(\Delta F, \Delta F), \tag{2.6}$$

which represents a local model to the objective function.

3. Trust-Region Method

The trust-region subproblem associated with the problem (1.4)–(1.5) is given by

$$\min_{\Delta F} \qquad q(\Delta F) \quad \text{subject to} \quad \|\Delta F\| \le \delta, \tag{3.1}$$

where $\delta > 0$ is the trust-region radius. By applying the first-order optimality conditions on the problem (3.1) we obtain the following result.

Theorem 3.1 *Let $F \in \mathcal{S}_F$, and let $P(F)$ and $S(F)$ be solutions to the discrete Lyapunov equations* (1.5) *and* (2.2)*, respectively. If ΔF solves* (3.1)*, then ΔF solves the linear matrix equation*

$$\begin{aligned} &(B^T S(F) B + R) \Delta F C P(F) C^T + B^T \Delta S(\Delta F) A(F) P(F) C^T \\ &\qquad + (B^T S(F) A(F) + RFC) \Delta P(\Delta F) C^T + \lambda \Delta F = -\nabla J(F), \end{aligned} \tag{3.2}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier associated with the trust-region constraint, $\Delta S(\Delta F)$ and $\Delta P(\Delta F)$ solve, respectively, (2.5) *and*

$$\Delta P(\Delta F) = A(F) \Delta P(\Delta F) A(F)^T + B \Delta F C P(F) A(F)^T + A(F) P(F) C^T \Delta F^T B^T, \tag{3.3}$$

and $\nabla J(F)$ is given by (2.3).

Proof. By forming the Lagrangian function associated with the trust-region constraint of (3.1) and differentiating that Lagrangian function with respect to ΔF we obtain the linear matrix equation (3.2) coupled with the discrete Lyapunov equations (2.5) and (3.3). □

Observe that because of using the directional derivatives an explicit form for the Hessian matrix of the quadratic model (2.6) is not available. Let us denote that Hessian by $\mathcal{H}$. Then (3.2) can be written in the compact form:

$$\mathcal{H}\Delta F + \lambda \Delta F = -\nabla J(F). \tag{3.4}$$

The trust-region algorithm for solving the optimization problem (1.4)–(1.5) is iterative. At the kth iteration our method solves a subproblem of the form (3.1) where the minimizer ΔF_k of the quadratic model $q_k(\Delta F)$ is trusted locally to approximate the objective of the problem (1.4)–(1.5) in a region of radius $\delta_k > 0$. The trust-region radius δ_k is updated every iteration. An efficient way of solving (3.1) is to compute an *approximate* solution to this subproblem specially when the size of the problem is large. This is done by using a modified Steihaug's conjugate gradient (CG) method; see, e.g., [2]. Thus we apply the CG to the linear system (3.2) (with $\lambda = 0$) coupled with the discrete Lyapunov equations (2.5), (3.3) where the computed step is restricted to satisfy the trust-region constraint. The only modification on Steihaug's algorithm is the supplemental restriction on the computed step ΔF to be such that $F_k + \Delta F \in \mathcal{S}_F$.

The CG algorithm is stated below. In this algorithm W represents the approximation of the Newton step ΔF while U and D are the residual and the direction required by the CG method, respectively.

Algorithm 3.1 (Conjugate gradient trust-region algorithm for solving (3.1))
Let $\delta_k > 0$ be given. Set $W := 0 \in \mathbb{R}^{n_u \times n_y}$, $U := -\nabla J(F_k)$, $D := U$, $\epsilon_{\text{cg}} = 0.01\,\|U\|$. Solve (2.5) and (3.3) for $\Delta S(D)$ and $\Delta P(D)$, respectively.
Repeat at most $n_u \times n_y$ times.

1. Compute $\mathcal{K} = \langle D, \mathcal{H}_k\, D\rangle$, the ratio $\alpha = \|U\|^2/\mathcal{K}$, and then set $W^+ = W + \alpha\, D$.
2. If $\mathcal{K} \le 0$ or $\|W^+\| > \delta_k$, then compute

$$\bar{\theta} := \max\left\{\theta > 0 \;:\; \|W + \theta\, D\| \le \delta_k,\; F_k + (W + \theta D) \in \mathcal{S}_F\right\}.$$

   Set $W^+ = W + \bar{\theta}\, D$, and stop.
3. Compute $U^+ = U - \alpha\, \mathcal{H}_k\, D$. If $\|U^+\| < \epsilon_{\text{cg}}$, stop.
4. Compute $\beta = \|U^+\|^2/\|U\|^2$ and $D^+ = U^+ + \beta\, D$.
5. Set $W \leftarrow W^+$, $U \leftarrow U^+$, $D \leftarrow D^+$, and go to Step 1.

End(repeat)

This algorithm contains three exit possibilities similar to Steihaug's algorithm. The first two exits occur when the computed step yields a non-positive curvature $\mathcal{K} \le 0$ (i.e., when $\mathcal{H}_k$ is not positive definite) or if $\|W^+\| > \delta_k$ (Step 2). In these two cases we do backtracking to compute the maximum $\theta = \bar{\theta} > 0$ such that

$$\|W + \theta\, D\| \le \delta_k, \quad F_k + (W + \theta D) \in \mathcal{S}_F,$$

and exit the CG loop. The third exit takes place if the residual becomes less than the prescribed accuracy (Step 3). The parameter $\bar{\theta}$ of Step 2 is computed as follows. Given δ_k, W and D we solve for $\theta > 0$ the scalar quadratic equation

$$\|W + \theta\, D\|^2 = \delta_k^2.$$

If the computed θ is such that $F_k + (W + \theta D) \in \mathcal{S}_F$, then we set $\bar{\theta} := \theta$, update the step and exit the CG method. Otherwise, we decrease θ in a backtracking loop until we reach $\bar{\theta} > 0$ such that $F_k + (W + \bar{\theta} D) \in \mathcal{S}_F$.

After evaluating the approximate step $\Delta F := W^+$ it remains to accept or reject that trial step and to update the trust-region radius δ_k. The quantities $Ared_k(\Delta F)$ and $Pred_k(\Delta F)$ of the *actual* and *predicted* reductions of the objective function are used to measure the progress made by the computed trial step ΔF toward optimality. These quantities are defined as:

$$Ared_k(\Delta F) = J(F_k) - J(F_k + \Delta F), \tag{3.5}$$

and

$$Pred_k(\Delta F) = -q_k(\Delta F).$$

The ratio $Ared_k(\Delta F)/Pred_k(\Delta F)$ measures the progress towards optimality. According to the value of this ratio the computed trial step ΔF is accepted or rejected and consequently δ_k is increased or decreased. The trust-region algorithm is stated in the following lines; see also [2]. This algorithm terminates if the following criterion is satisfied:

$$\|\nabla J(F_k)\| \leq \epsilon_1^{\text{tol}}, \tag{3.6}$$

where $\epsilon_1^{\text{tol}} > 0$ is the tolerance.

Algorithm 3.2 (disTR: Trust-region method)
Let $F_0 \in \mathcal{S}_F$, $\delta_0 > 0$ and $\epsilon_1^{\text{tol}} \in (0, 1)$ be given. Solve (1.5) and (2.2) for $P(F_0)$ and $S(F_0)$, respectively. Choose $0 < \mu < \eta < 1$ and $0 < \gamma_1 < \gamma_2 < 1 < \gamma_3$. Set $k := 0$.
While $\|\nabla J(F_k)\| > \epsilon_1^{\text{tol}}$, do

1. Use Algorithm 3.1 to compute an approximate solution ΔF to (3.2) coupled with the discrete Lyapunov equations (2.5) and (3.3) such that $F_k + \Delta F \in \mathcal{S}_F$.
2. Compute the ratio $Ared_k(\Delta F)/Pred_k(\Delta F)$.
3. If $Ared_k(\Delta F)/Pred_k(\Delta F) < \mu$,
    set $\delta_{k+1} \in [\gamma_1 \delta_k, \gamma_2 \delta_k]$ and $F_{k+1} = F_k$.
   Else if $\mu \leq Ared_k(\Delta F)/Pred_k(\Delta F) < \eta$,
    set $\delta_{k+1} \in [\gamma_2 \delta_k, \delta_k]$ and $F_{k+1} = F_k + \Delta F$.
   Else
    set $\delta_{k+1} \in [\delta_k, \gamma_3 \delta_k]$ and $F_{k+1} = F_k + \Delta F$.
   End (if)
4. Solve (1.5) and (2.2) for $P(F_{k+1})$ and $S(F_{k+1})$, respectively.
5. Compute $J(F_{k+1})$ and $\|\nabla J(F_{k+1})\|$, and set $k \leftarrow k + 1$.

End(while)

Step 3 of Algorithm 3.2 holds the update rule of the trust-region radius δ_k and also provides the decision of accepting or rejecting the computed trial step ΔF. The trial step is rejected if $Ared_k(\Delta F)/Pred_k(\Delta F) < \mu$; otherwise it is accepted. According to the value assigned to the parameters μ and η the trust-region procedure decreases or increases δ_k. Observe that when the first *"If"* in Step 3 is encountered we decrease δ_k. On the other hand, δ_k is enlarged if the last *"Else"* is encountered. Finally, δ_k is left almost unchanged when the "*Else if*" lying in the middle of the if statement is encountered. In the implementation the following values have been assigned to the parameters in Algorithm 3.2: $\mu = 0.1$, $\eta = 0.3$, $\gamma_1 = 0.3, \gamma_2 = 0.8$, and $\gamma_3 = 2$. The initial trust-region radius is chosen as $\delta_0 = \|\nabla J(F_0)\|$. Note that in the literature of the trust-region methods there are various heuristics for updating the trust-region radius; see, e.g., [2].

4. Computing a Suboptimal Discrete Stabilizing Output Feedback Gain

By taking a look at Algorithm 3.2 we see that it requires an initial stabilizing output feedback controller, i.e., $F_0 \in \mathbb{R}^{n_u \times n_y}$ such that $\rho(A + BF_0C) < 1$. The aim of this section is to develop a first-order method for computing a suboptimal stabilizing output feedback controller that can be used to find $F_0 \in \mathcal{S}_F$. To achieve this goal we consider the following modified formulation of the optimization problem (1.4)–(1.5):

$$\min_{(F,\nu)\in\mathcal{S}_F^\nu} \quad J(F,\nu) = \mathrm{Tr}\,(P(F,\nu)Q(F)) + \sigma\nu^2, \tag{4.1}$$

where $P(F,\nu)$ solves the following discrete Lyapunov equation:

$$P = \bar{A}(F,\nu)P\bar{A}(F,\nu)^T + V, \tag{4.2}$$

$Q(F) := Q + C^TF^TRFC$, $\bar{A}(F,\nu) := (1-\nu)A + BFC$, $\nu \in [0,1)$, $V := \mathbb{E}\{x_0x_0^T\}$ is the covariance matrix, σ is a large positive constant (in the implementation $\sigma = 10^7$), and

$$\mathcal{S}_F^\nu := \{(F,\nu) \in \mathbb{R}^{n_u\times n_y} \times \mathbb{R}_+ \ : \ \rho(\bar{A}(F,\nu)) < 1\} \tag{4.3}$$

is the perturbed set of stabilizing output feedback controllers. The constant matrix V can be chosen to be the identity or any symmetric positive definite matrix. Observe that in (4.1)–(4.3) we parameterize the constant system matrix A and replace it by $A_\nu = (1-\nu)A$, where $\nu \in [0,1)$ is a scalar variable; see, e.g., [4].

Larin in [4] has considered the idea of replacing the matrix A by A_ν and including ν as a quadratic penalty term in the objective function. Moreover, he obtained first-order derivatives of the objective function. However, he did not develop any computational method for solving the modified penalized problem (4.1)–(4.2); he only suggested to use the function fminu.m from the optimization toolbox of MATLAB to solve that problem. In fact using the function fminu.m directly as was suggested in [4] is not a good choice since there is no guarantee that the obtained solution stabilizes the associated closed-loop system matrix $\bar{A}(F,\nu)$.

In the following we attempt to obtain a stationary point of the modified problem (4.1)–(4.2) iteratively by solving the corresponding nonlinear system of matrix equations that results from the first-order necessary optimality conditions. The choice $(F_0,\nu_0) = (0,\nu_0)$ will be our spontaneous choice to start the method such that $\rho(A_{\nu_0}) < 1$.

First-order derivatives of $J(F,\nu)$ are given in the following lemma.

Lemma 4.1 *The first-order directional derivatives of $J(F,\nu)$ in the direction of ΔF and $\Delta\nu$ are given by*

$$J_F(F,\nu)\Delta F \;=\; 2\,\mathrm{Tr}\,((B^TS(F,\nu)\bar{A}(F,\nu) + RFC)P(F,\nu)C^T\Delta F^T), \tag{4.4}$$

$$J_\nu(F,\nu)\Delta\nu \;=\; 2\,[\sigma\nu - \mathrm{Tr}\,(\bar{A}(F,\nu)^TQ(F)AP(F,\nu))]\Delta\nu, \tag{4.5}$$

where $\bar{A}(F,\nu) = A_\nu + BFC$, and $P(F,\nu)$ and $S(F,\nu)$ are given, respectively, as solutions of the discrete Lyapunov equations (4.2) *and*

$$S = \bar{A}(F,\nu)^TS\bar{A}(F,\nu) + Q(F). \tag{4.6}$$

Proof. The first relation is similar to (2.1); see, e.g., [12]. On the other hand, by using (4.2) and differentiating the objective function with respect to the scalar variable ν we obtain

$$\begin{aligned}\partial\,\mathrm{Tr}\,(P(F,\nu)Q(F))/\partial\nu &= \mathrm{Tr}\,((\partial P(F,\nu)/\partial\nu)Q(F)) \\ &= -2\,\mathrm{Tr}\,(\bar{A}(F,\nu)P(F,\nu)A^TQ(F)) \\ &= -2\,\mathrm{Tr}\,(\bar{A}(F,\nu)^TQ(F)AP(F,\nu)),\end{aligned}$$

where $\partial Q(F)/\partial \nu = 0$, $\partial V/\partial \nu = 0$, and $\partial \bar{A}(F,\nu)/\partial \nu = -A$. Consequently, the directional derivative of J with respect to ν yields (4.5). □

To simplify the notations let us write P and S without their argument (F,ν). The first-order optimality conditions of the modified optimization problem (4.1)–(4.2) are

$$\nabla_F J(F,\nu) \equiv 2\,(B^T S\bar{A}(F,\nu) + RFC)PC^T \;=\; 0, \tag{4.7}$$

$$\nabla_\nu J(F,\nu) \equiv 2\,(\sigma\nu - \mathrm{Tr}\,(\bar{A}(F,\nu)^T Q(F)AP)) \;=\; 0, \tag{4.8}$$

where P and S solve the discrete Lyapunov equations (4.2) and (4.6), respectively.

Note that (4.8) yields ν explicitly in terms of F and P:

$$\nu = \mathrm{Tr}\,(A(F)^T Q(F)AP)/(\sigma + \mathrm{Tr}\,(A^T Q(F)AP)), \tag{4.9}$$

where $A(F) = \bar{A}(F,0)$.

Let us consider the following assumption required in constructing the method.

Assumption 4.1 *Assume that the given weight matrix R is positive definite and the constant matrix C has full row rank. Assume further that the matrix variable P is positive definite.*

The above assumption is classical and can be found, e.g., in [9]. In the implementation the weight matrix R is taken as the identity or any positive definite matrix. Lyapunov stability theory guarantees that if V is positive definite then so is the solution P of the discrete Lyapunov equation (4.2) whenever $\bar{A}(F,\nu)$ is a stability matrix.

Lemma 4.2 *Under Assumption 4.1 the gradient equation* (4.7) *gives*

$$F = -R^{-1}B^T S\bar{A}(F,\nu)PC^T(CPC^T)^{-1}. \tag{4.10}$$

Proof. Since by assumption R^{-1} and $(CPC^T)^{-1}$ exist, (4.10) follows from (4.7). □

We now describe the computational method for solving the system of equations (4.7), (4.9), (4.2), and (4.6) iteratively. This method is to some extent similar to the Algorithm of Levine and Athans [9]. Initializing the method is done as follows. First, we set $F_0 = 0$ and choose $\nu_0 \in (0,1)$ such that $\rho(A_{\nu_0}) < 1$. Then by solving the discrete Lyapunov equations (4.2) and (4.6) we obtain P_0 and S_0, respectively. In order to describe a general iteration let us denote the current and the new iteration variables as (F,P,S,ν) and $(\widetilde{F},\widetilde{P},\widetilde{S},\tilde{\nu})$, respectively. Given F,P,S and ν an update $\widetilde{F}$ of F is computed by the right-hand-side of (4.10). For given $\widetilde{F}$ and ν we solve the discrete Lyapunov equation (4.2):

$$P = \bar{A}(\widetilde{F},\nu)P\bar{A}(\widetilde{F},\nu)^T + V. \tag{4.11}$$

Let $\widetilde{P}(\widetilde{F},\nu)$ be the corresponding solution. Next, (4.6) can be rewritten as

$$-S + (A_\nu + BFC)^T S(A_\nu + BFC) + C^T F^T RFC + Q = 0. \tag{4.12}$$

By substituting the right-hand-side of (4.10) into (4.12) we have the following nonlinear matrix equation:

$$\begin{aligned} -S \;\; + \;\; & M(F,P,S,\nu)^T SM(F,P,S,\nu) + C^T(CPC^T)^{-1}(B^T S\bar{A}(F,\nu)PC^T)^T R^{-1} \\ & \times(B^T S\bar{A}(F,\nu)PC^T)(CPC^T)^{-1}C + Q = 0, \end{aligned} \tag{4.13}$$

where

$$M(F,P,S,\nu) = A_\nu - BR^{-1}B^T S\bar{A}(F,\nu)PC^T(CPC^T)^{-1}C.$$

Obviously (4.13) is nonlinear in S and can be solved by any nonlinear solver, e.g., the function fsolve from the optimization toolbox of MATLAB. Let $\widetilde{S}$ be the corresponding solution. Finally, for given $\widetilde{F}$ and $\widetilde{P}$ we compute a new estimate $\tilde{\nu}$ for ν by using (4.9).

A reasonable stopping criterion for this procedure is the following:

$$\|\nabla_F J(F,\nu)\| + \|\nabla_\nu J(F,\nu)\| \le \epsilon_2^{\mathrm{tol}}, \tag{4.14}$$

where $\epsilon_2^{\mathrm{tol}}$ is the tolerance. Another possible stopping criterion might be

$$\max\Big\{\|\nabla_F J(F,\nu)\|,\ \|\nabla_\nu J(F,\nu)\|\Big\} \le \epsilon_2^{\mathrm{tol}}. \tag{4.15}$$

The proposed algorithm is stated in the following lines.

Algorithm 4.1 (Fstab:First-order method for computing suboptimal stabilizing controller) Let $\sigma > 0$ be a very large number and let $\epsilon_2^{\mathrm{tol}} \in (0,1)$ be given. Let $F_0 = 0$ and $\nu_0 \in (0,1)$ satisfying $\rho(A_{\nu_0}) < 1$. Choose $0 < \epsilon < \alpha \le \beta < 1$. Compute P_0 and S_0 solutions of the discrete Lyapunov equations (4.2) and (4.6), respectively. Set $k := 0$.

While the termination criterion (4.14) (or (4.15)) is not achieved, do

1. Compute F_{k+1} by the right-hand-side of (4.10) with F_k, P_k, S_k and ν_k. Set $j := 0$.
   While $(F_{k+1}, \nu_k) \notin \mathcal{S}_F^\nu$, do (perform backtracking)
   (a) Set $F_{k+1} = F_k + \beta\alpha^j(F_{k+1} - F_k)$.
   (b) If $\beta\alpha^j \le \epsilon$, stop; otherwise set $j \leftarrow j+1$.
   End(while)
2. Given F_{k+1} and ν_k solve the discrete Lyapunov equation (4.11) for P_{k+1}.
3. Given F_{k+1}, P_{k+1} and ν_k solve the nonlinear matrix equation (4.13) for S_{k+1}.
4. Compute $\nu_{k+1} = \nu(F_{k+1}, P_{k+1})$ by using (4.9), and set $k \leftarrow k+1$.

End(while)

Observe that since we are seeking a suboptimal stabilizing output feedback controller it is not necessary to execute the method Fstab until the convergence criterion reaches a very small value to the tolerance $\epsilon_2^{\mathrm{tol}}$. It is sufficient to execute the outer loop two or three iterations to obtain an F such that $(F,\nu) \in \mathcal{S}_F^\nu$ with ν sufficiently small. In fact the large value assigned to the constant σ produces a very fast reduction in ν in early iterations.

Algorithm 4.1 can be modified by exchanging Steps 2 and 3 as follows:

2. Given F_{k+1}, P_k and ν_k solve the nonlinear matrix equation (4.13) for S_{k+1}.
3. Given F_{k+1} and ν_k solve the discrete Lyapunov equation (4.11) for P_{k+1}.

In the implementation the following values have been assigned to the parameters in Algorithm 4.1: $\alpha = 0.5$, $\beta = 0.8$ and $\epsilon = 1 \times 10^{-5}$.

Let $(F_s, \nu_s) \in \mathcal{S}_F^\nu$ be the final iterate computed by Fstab such that ν_s is negligible. Since we can expect that the obtained F_s usually lies in $\mathcal{S}_F$ or at least we can view it as to be approximately in $\mathcal{S}_F$, it will be used to start the method disTR.

5. Numerical Results

In this section an implementation for the method disTR is described. A MATLAB code was written corresponding to this implementation. For the sake of comparison we have implemented Newton's method with Armijo stepsize rule as introduced in [12], denoted by Newton–Armijo. On the other hand, several discrete Lyapunov equations have to be solved every iteration. The MATLAB function dlyap$(\cdot,\cdot)$ is employed to compute these solutions.

In addition, the method Fstab is also implemented to determine an initial stabilizing output feedback gain matrix required to start the method disTR if the considered uncontrolled system is not discrete-time Schur stable[1] .

In the following we consider five test problems from the literature in detail which can quite demonstrate the performance of the method disTR. For each example we list some of the obtained data in tables. In these tables the first to the fifth columns are, respectively, the iteration counter k, the objective function $J(F_k)$, the convergence criterion $\|\nabla J(F_k)\|$, the trust-region radius δ_k, and the number of CG iterations i_{cg} required in solving the linear matrix equation (3.2).

For all test problems the tolerance of (3.6) is chosen to be $\epsilon_1^{\text{tol}} = 1 \times 10^{-7}$.

Example 5.1 This test problem describes a one-input one-output discrete-time model (see [16]). The data matrices are as follows:

$$A = \begin{bmatrix} 0.5477 & 0.8208 & 0 \\ -0.8208 & 0.5067 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, C^T = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

while the weight matrices Q, R, and the covariance matrix V were chosen as follows:

$$Q = 100\, I_3, \qquad R = 1.5\, I_1, \qquad V = 0.8\, I_3.$$

The uncontrolled system is discrete-time Schur stable. Therefore, we can start with $F_0 = 0 \in \mathbb{R}^{1\times 1}$. Our main task is to compute a *stabilizing* static output feedback gain matrix $F \in \mathcal{S}_F$ that solves the optimal control problem (1.1) which equivalently solves the unconstrained optimization problem (1.4)–(1.5).

Table 1: Performance of the method disTR on Example 5.1

k	$J(F_k)$	$\|\nabla J(F_k)\|$	δ_k	i_{cg}
1	2.4204e+003	7.2136e+003	2.7366e+004	1
2	1.7142e+003	2.9654e+003	5.4731e+004	1
3	1.2675e+003	1.1734e+003	1.0946e+005	1
⋮	⋮	⋮	⋮	⋮
8	8.0685e+002	8.8244e−002	3.5028e+006	1
9	8.0685e+002	2.8753e−005	7.0056e+006	1
10	8.0685e+002	3.2407e−012	1.4011e+007	1

By applying Newton–Armijo method and the proposed method disTR on this test problem we obtain the following results. Newton–Armijo method converges to the stationary point F_* in 11 iterations while the method disTR reaches the same stationary point F_* in 10 iterations. The computed discrete static output feedback gain is

$$F_* = -0.8505.$$

Table 1 shows the convergence behavior of the method disTR. We see that starting from a remote point $F_0 \in \mathcal{S}_F$ the method disTR converges to F_* with fast local rate of convergence.

[1]A matrix is called discrete-time Schur stable if its spectral radius is strictly less than one.

Example 5.2 The second test problem describes a simulated hydraulic turbine model (see [20]). The linearized model of the discrete dynamical system has the following data matrices:

$$A = \begin{bmatrix} 0.0067 & 0 & 0 \\ 0.0590 & 0.9875 & 0.0331 \\ 1.6359 & -0.0022 & 0.7846 \end{bmatrix}, B = \begin{bmatrix} 0.9933 \\ -0.0341 \\ -1.6315 \end{bmatrix}, C^T = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and the matrices Q, R, V were chosen as follows:

$$Q = 100\, I_3, \qquad R = 1.5\, I_1, \qquad V = 0.8\, I_3.$$

The uncontrolled system is discrete-time Schur stable. Therefore, we have started with $F_0 = 0 \in \mathbb{R}^{1\times 2}$.

Newton–Armijo method could not converge for this example; the convergence criterion stacks at the value 3.1376e−006. This happens because the method encounters a direction which is not descent. On the other hand, the solver disTR reaches the stationary point F_* in 6 iterations. Table 2 shows the convergence behavior of the method disTR for this example.

Table 2: Performance of the method disTR on Example 5.2

k	$J(F_k)$	$\|\nabla J(F_k)\|$	δ_k	$i_{\rm cg}$
1	3.6940e+003	8.4913e+002	4.6309e+003	1
2	3.5100e+003	2.2804e+002	9.2619e+003	1
3	3.4695e+003	3.9003e+001	1.8524e+004	1
4	3.4685e+003	9.1695e−002	3.7048e+004	2
5	3.4685e+003	3.8273e−006	7.4095e+004	2
6	3.4685e+003	3.2924e−011	1.4819e+005	2

The computed discrete static output feedback gain F_* is

$$F_* = [\, -0.2551 \quad 0.1602 \,].$$

Example 5.3 This example is the discrete model of a BOEING B–747 aircraft model (see [5, AC5]). The discrete linearized dynamical system for this model has the following data matrices:

$$A = \begin{bmatrix} 0.9801 & 0.0003 & -0.0980 & 0.0038 \\ -0.3868 & 0.9071 & 0.0471 & -0.0008 \\ 0.1591 & -0.0015 & 0.9691 & 0.0003 \\ -0.0198 & 0.0958 & 0.0021 & 1 \end{bmatrix}, B = \begin{bmatrix} -0.0001 & 0.0058 \\ 0.0296 & 0.0153 \\ 0.0012 & -0.0908 \\ 0.0015 & 0.0008 \end{bmatrix}, C^T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$Q = V = I_4$ and $R = I_2$.

The uncontrolled system in this example is also discrete-time Schur stable. Therefore, we are able to start the method with $F_0 = 0 \in \mathbb{R}^{2\times 2}$.

Newton–Armijo method converges to the optimal discrete static output feedback gain matrix F_* in 15 iterations while disTR needs 13 iterations to achieve the same point:

$$F_* = \begin{bmatrix} 1.4057 & -0.6857 \\ -1.1432 & 0.0015 \end{bmatrix}.$$

Table 3 shows the convergence behavior of the method disTR on this test problem while Figure 1 exhibits the effect of the computed discrete-time static output feedback gain F_* on the control system of (1.1). Although the uncontrolled system is stable the computed output feedback gain matrix F_* causes all state variables to be steered to the zero state faster than the case of uncontrolled system.

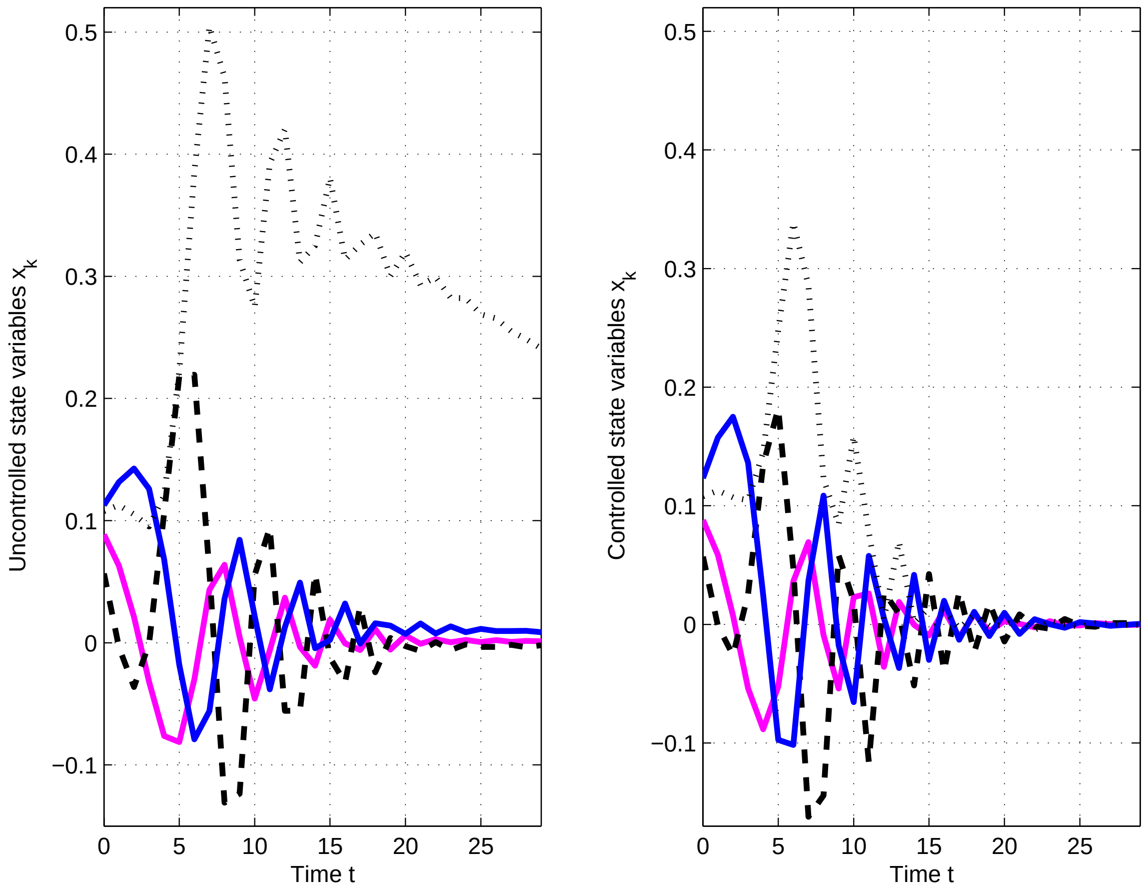

Figure 1: Uncontrolled and controlled state space models for Example 5.3

Example 5.4 This test problem is obtained from the benchmark collection [5, DIS5]. It has the following data matrices:

$$A = \begin{bmatrix} 0.8189 & 0.0863 & 0.0900 & 0.0813 \\ 0.2524 & 1.0033 & 0.0313 & 0.2004 \\ -0.0545 & 0.0102 & 0.7901 & -0.2580 \\ -0.1918 & -0.1034 & 0.1602 & 0.8604 \end{bmatrix}, B = \begin{bmatrix} 0.0045 & 0.0044 \\ 0.1001 & 0.0100 \\ 0.0003 & -0.0136 \\ -0.0051 & 0.0936 \end{bmatrix}, C^T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},$$

$Q = V = I_4$ and $R = I_2$.

The uncontrolled system is not discrete-time Schur stable; $\rho(A) = 1.0192 > 1$. Therefore, we have used the method Fstab to compute an $F_0 \in \mathcal{S}_F$. The method Fstab was initiated by $(F_0, \nu_0) = (0, 0.1)$ such that $\rho(\bar{A}(0, 0.1)) = 0.9173 < 1$. After 2 iterations Fstab reaches the stabilizing pair

$$F_s = \begin{bmatrix} -0.7963 & -0.2130 \\ -0.1514 & -0.0489 \end{bmatrix}, \qquad \nu_s = 1.5632\text{e}{-007},$$

where $\rho(\bar{A}(F_s, \nu_s)) = 0.9720 < 1$. The corresponding value of the objective function is $J_s = 7.0795\text{e}{+}001$. Observe that we do not require the convergence criterion in the method Fstab to reach a very small value. It is sufficient for Fstab to achieve a stabilizing output feedback controller $F_s \in \mathcal{S}_F$ with ν_s being negligible. By choosing the obtained F_s as an initial iterate to both of Newton–Armijo and disTR the two methods give the following results. Newton–Armijo method converges to the optimal output feedback controller F_* in

8 iterations while disTR needs 7 iterations to achieve the same stationary point:

$$F_* = \begin{bmatrix} -1.5802 & -0.2700 \\ -0.2348 & -0.0428 \end{bmatrix}.$$

Table 4 shows the convergence behavior of the method disTR on this test problem. Obviously from Table 4 the final value of the objective function $J(F_*) = 5.2626\text{e}{+}001$ is strictly less than J_s that corresponds to F_s computed by the method Fstab.

Table 4: Performance of the method disTR on Example 5.4

k	$J(F_k)$	$\|\nabla J(F_k)\|$	δ_k	i_{cg}
1	6.0215e+001	1.1355e+001	9.0205e+001	1
2	5.5775e+001	3.5921e+000	1.8041e+002	1
3	5.2892e+001	7.8423e−001	3.6082e+002	2
4	5.2647e+001	1.5241e−001	7.2164e+002	2
5	5.2626e+001	5.7622e−003	1.4433e+003	3
6	5.2626e+001	2.5723e−006	2.8865e+003	3
7	5.2626e+001	8.2974e−013	5.7731e+003	4

Example 5.5 This test problem is from [17] of a linearized discrete-system model and has the following data matrices:

$$A = \begin{bmatrix} 0.2113 & 0.0087 & 0.4524 \\ 0.0824 & 0.8096 & 0.8075 \\ 0.7599 & 0.8474 & 0.4832 \end{bmatrix}, B = \begin{bmatrix} 0.6135 & 0.6538 \\ 0.2749 & 0.4899 \\ 0.8807 & 0.7741 \end{bmatrix}, C = Q = V = I_3, \quad R = I_2.$$

The uncontrolled system is not discrete-time Schur stable ($\rho(A) = 1.6133 > 1$). Therefore, we have used the method Fstab to compute $F_0 \in \mathcal{S}_F$. The method Fstab was started by $(F_0, \nu_0) = (0, 0.65)$ such that $\rho(\bar{A}(0, 0.65)) = 0.5647 < 1$. After 4 iterations Fstab reaches the stabilizing pair:

$$F_s = \begin{bmatrix} -0.6134 & -0.5299 & -0.5401 \\ -0.4773 & -0.5999 & -0.8850 \end{bmatrix}, \qquad \nu_s = 5.5215\text{e}{-}007,$$

where $\rho(\bar{A}(F_s, \nu_s)) = 0.8974 < 1$. The corresponding value of the objective function is $J_s = 7.2241\text{e}{+}002$. By using F_s as initial iterate to both of Newton–Armijo and disTR the two methods yield the following results. Newton–Armijo method stacks at the value 1.6351e−007 which is very close to the tolerance of value 1.0e−007. Similar to Example 2 the method encounters an ascent direction that prevents the method from convergence. On the other hand, disTR needs 10 iterations to reach the stationary point:

$$F_* = \begin{bmatrix} -1.1139 & 0.4723 & 1.1186 \\ 0.4554 & -1.3619 & -1.9418 \end{bmatrix}.$$

Table 5 shows the convergence behavior of the method disTR on this test problem. The final value of the objective function $J(F_*)$=3.0070e+002 is clearly less than J_s that corresponds to F_s computed by the method Fstab.

In addition to the above state feedback case we have also chosen the matrix C to be

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Table 5: Performance of the method disTR on Example 5.5

k	$J(F_k)$	$\|\nabla J(F_k)\|$	δ_k	$i_{\rm cg}$
1	5.6997e+002	6.7025e+002	6.0339e+003	1
2	4.6900e+002	3.0120e+002	1.2068e+004	1
3	4.0303e+002	1.3495e+002	2.4136e+004	1
⋮	⋮	⋮	⋮	⋮
8	3.0070e+002	2.0611e−002	7.7234e+005	4
9	3.0070e+002	1.3908e−004	1.5447e+006	5
10	3.0070e+002	4.1671e−008	3.0894e+006	5

In this case the method Fstab achieves after 2 iterations the following pair:

$$F_s = \begin{bmatrix} -0.3443 & -0.4099 \\ -0.3217 & -0.4454 \end{bmatrix}, \qquad \nu_s = 3.6529\text{e}{-006},$$

where $\rho(\bar{A}(F_s, \nu_s)) = 0.7028 < 1$. The corresponding value of the objective function is $J_s = 6.2098\text{e}{+002}$. Choosing F_s as a starting point Newton–Armijo method converges to the stationary point

$$F_* = \begin{bmatrix} -1.3219 & 0.5384 \\ 0.5817 & -1.7087 \end{bmatrix},$$

after 9 iterations while disTR needs 8 iterations to reach the same point. The corresponding value of the objective function is $J(F_*) = 4.5147\text{e}{+002}$.

The main conclusion that we can draw from the above results is that the method disTR outperforms Newton–Armijo method [12] on the considered set of discrete test problems with respect to number of iterations. The above tables show the fast local rate of convergence of disTR starting from any remote point $F_0 \in \mathcal{S}_F$.

On the other hand, the proposed first-order method Fstab has been applied to compute suboptimal discrete stabilizing output feedback controllers. These suboptimal output feedback controllers are used to initiate the method disTR on the fourth and fifth test problems where the corresponding control systems are not discrete-time Schur stable.

Acknowledgements. The author would like to thank the Associate Editor and an anonymous referee for their careful reading and for their helpful comments.

References

[1] S.P. Boyd, L. El-Ghaoui, E. Feron and V. Balakrishnan: *Linear Matrix Inequalities in System and Control Theory, Studies in Applied Mathematics 15* (SIAM, Philadelphia, 1994).

[2] A.R. Conn, N.I.M. Gould and Ph.L. Toint: *Trust-Region Methods, MPS-SIAM Series on Optimization 1* (SIAM, Philadelphia, 2000).

[3] T. Iwasaki, R.E. Skelton and J.C. Geromel: Linear quadratic suboptimal control with static output feedback. *Systems & Control Letters*, **23** (1994), 421–430.

[4] V.B. Larin: Stabilization of the system by static output feedback. *Applied and Computational Mathematics*, **2** (2003), 2–12.

[5] F. Leibfritz: COMP*lib*: COnstraint Matrix-optimization Problem *lib*rary—a collection of test examples for nonlinear semi-definite programs, control system design and related problems. Technical Report 2004. `http://www.complib.de/`

[6] F. Leibfritz: An LMI-based algorithm for designing suboptimal static $\mathcal{H}_2/\mathcal{H}_\infty$ output feedback controllers. *SIAM Journal on Control and Optimization*, **39** (2001), 1711–1735.

[7] F. Leibfritz and E.M.E. Mostafa: Trust region methods for solving the optimal output feedback design problem. *International Journal of Control*, **76** (2003), 501–519.

[8] F. Leibfritz and E.M.E. Mostafa: An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems. *SIAM Journal on Optimization*, **12** (2002), 1048–1074.

[9] W.S. Levine and M. Athans: On the determination of the optimal constant output feedback gains for linear multivariable systems. *IEEE Transactions on Automatic Control*, **15** (1970), 44–48.

[10] Y.-C. Lin and J.-C. Lo: $\mathcal{H}_2$ suboptimal fuzzy control via dynamic output feedback for continuous-time systems. Technical Report, Dept. of Mechanical Engineering, National Central University, Chung-Li, Taiwan.

[11] W.Q. Liu and V. Sreeram: New algorithm for computing LQ suboptimal output feedback gains of decentralized control systems. *Journal of Optimization Theory and Applications*, **93** (1997), 597–607.

[12] P.M. Mäkilä and H.T. Toivonen: Computational methods for parametric LQ problems–a survey. *IEEE Transactions on Automatic Control*, **32** (1987), 658–671.

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

[14] E.M.E. Mostafa: A trust region method for solving the decentralized static output feedback design problem. *Journal of Applied Mathematics & Computing*, **18** (2005), 1–23.

[15] E.M.E. Mostafa: An augmented Lagrangian SQP method for solving some special class of nonlinear semi-definite programming problems. *Computational and Applied Mathematics*, **24** (2005), 461–486.

[16] G. Pannocchia, S.J. Wright and J.B. Rawlings: Existence and computation of infinite horizon model predictive control with active steady-state constraints. *IEEE Transactions on Automatic Control*, **48** (2003), 1002–1006.

[17] P.L.D. Peres and J.C. Geromel: $\mathcal{H}_2$ control for discrete-time systems optimality and robustness. *Automatica*, **29** (1993), 225–228.

[18] J.-K. Shiau and J.H. Chow: Structurally constrained $\mathcal{H}_\infty$ suboptimal control design using an iterative linear matrix inequality algorithm based on a dual design formulation. *Tamkang Journal of Science and Engineering*, **1** (1998), 133–143.

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

[20] H. Wang and S. Daley: A fault detection method for unknown systems with unknown input and its application to hydraulic turbine monitoring. *International Journal of Control*, **57** (1993), 247–260.

El–Sayed M.E. Mostafa
Dept. of Mathematics
Faculty of Science
Alexandria University
Alexandria, Egypt
E-mail: `emostafa99@yahoo.com`