IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 17, Issue 3 Ser. II (May – June 2021), PP 12-26
www.iosrjournals.org

A Hybrid Conjugate Gradient Method for Discrete–Time Periodic Static Output Feedback Control Design

[1]Fatma F.S. Omar
(Mathematics Department/University of Gharyan, Libya)

Abstract: *In this paper, the discrete–time periodic static output feedback control design problem is considered. A hybrid conjugate gradient methods are analyzed and studied to tackle an equivalent optimization problem of this optimal control problem. Finally, the proposed algorithms are tested numerically through several test problems from the benchmark collection.*

Key words: *Output feedback control, Discrete-time periodic system, Hybrid conjugate gradient methods.*

Date of Submission: 28-05-2021 Date of Acceptance: 10-06-2021

I. Introduction

The static output feedback design problem for discrete or continuous–time control systems is one of the most studied problems, where wide area of applications in engineering and in finance are represented by this problem; see the two surveys [19, 27] and the references therein. Particularly, many special purpose methods are designed by the engineers for solving this problem; see e.g. [19, 27]. Various gradient–based methods are available for solving the SOF problem among them is the descent Anderson–Moore method [19] that solves the SOF problem by minimizing particular quadratic approximation of the objective function combined with step–size rule. Mäkilä and Toivonen [19] solved the discrete problem by Newton's method with line search globalization. Rautert and Sachs [26] suggested quasi–Newton method with line search for solving the continuous–time SOF problem. Mostafa [21] introduced trust region method for solving the discrete–time SOF problem. All these methods are based on reformulating the discrete or continuous–time SOF problems into unconstrained matrix optimization problems. The formulation of the SOF problem as a constrained optimization problem allows utilizing numerous available constrained optimization techniques; see e.g [18, 17, 13, 22, 23]. In this paper hybrid conjugate gradient (HCG) methods are analyzed for tackling the SOF design problem. The proposed algorithm is extended for solving the design problem for periodic–time systems. The considered approach is based on reformulating the optimal control problem into an optimization problem. This approach is classical and we consider it for the sack of completeness.

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 $\langle M_1, M_2\rangle = Tr(M_1^T, M_2)$ for $M_1, M_2 \in \mathbb{R}^{m\times m}$ and $Tr(\cdot)$ is the trace operator. I_m denotes the $m \times m$ identity matrix.

II. The discrete–time periodic SOF design problem

Periodic–time control systems have been studied recently in several research articles in particular for the stabilization of systems of walking and hopping robots, see e.g, among others [1, 3, 4, 8, 29]. Consider the following problem of designing a stabilizing static output feedback controller of linear periodic discrete-time systems; see [1]:

$$\min \quad J(K_t) = E\{\textstyle\sum_{t=0}^{\infty} (x_t^T Q_t x_t + u_t^T R_t u_t)\} = E\{\textstyle\sum_{t=0}^{\infty} x_t \overline{Q}_t x_t\}, \qquad (1)$$

$$\text{s.}t. \quad x_{t+1} = A_t x_t + B_t u_t, \quad y_t = C_t x_t, \qquad (2)$$

where $E\{.\}$ is the expected value and $\overline{Q}_t := Q_t + C_t^T K_t^T R_t K_t C_t$, $x_t \in \mathbb{R}^m$, $u_t \in \mathbb{R}^p$ and $y_t \in \mathbb{R}^r$ are the state, the control input, and the measured output vectors. Moreover, $Q_t \in \mathbb{R}^{m\times m}, R_t \in \mathbb{R}^{p\times p}$ are given symmetric, periodic and positive semi–definite, positive definite weight matrices, respectively; $A_t \in \mathbb{R}^{m\times m}, B_t \in \mathbb{R}^{p\times p}$ and $C_t \in \mathbb{R}^{r\times m}$ are given d–periodic matrices, i.e.,

$$Q_{t+d} = Q_t, \quad R_{t+d} = R_t, \quad A_{t+d} = A_t, \quad B_{t+d} = B_t, \quad C_{t+d} = C_t, \quad \forall\ t, \quad d \geq 1.$$

We consider the following control law to close the system (1) – (2) :

$$u_t = K_t y_t, \qquad (3)$$

---

[1] Department of Mathematics, Faculty of Science, Gharyan University, Gharyan, Libya (fatma.omar@gu.edu.ly, ffsaidm@yahoo.com)

where $K_{t+d} = K_t$ for all t and $d \geq 1$. This yields the recurrence relation:

$$x_{t+1} = (A_t + B_t K_t C_t)\, x_t := \overline{A}_t x_t,$$

where

$$\overline{A}_t = A_t + B_t K_t C_t, \qquad \overline{A}_{t+d} = \overline{A}_t, \qquad t = 0,1,2,\dots .$$

We assume for simplicity that $d = 2$, from Lemma (6.2) in the paper [24], the system (2) is equivalently rewritten in the following closed form:

$$x_{t+1} = \overline{A}_1 \overline{A}_0 x_t = \psi(K_0, K_1) x_t,$$

where

$$\psi(K_0, K_1) = \overline{A}_1 \overline{A}_0. \tag{4}$$

From Lyapunov stability theory the matrix variables K_0 and K_1 must be chosen from the following set of stabilizing output feedback gains:

$$\mathcal{D}_d = \{K_0, K_1 \in \mathbb{R}^{p\times r} : \rho(\psi(K_0, K_1)) < 1\}. \tag{5}$$

Such a restriction ensures that all state variables decay to zero state as t increases.
Let us consider the sum of the first $d = 2$ terms of the following formula:

$$\textstyle\sum_{l=0}^{1} x_0^T (\overline{Q}_0 + \overline{A}_0^T \overline{Q}_1 \overline{A}_0) x_0 = \sum_{l=0}^{1} x_0^T L(K_0, K_1) x_0, \tag{6}$$

where

$$L(K_0, K_1) = \overline{Q}_0 + \overline{A}_0^T \overline{Q}_1 \overline{A}_0. \tag{7}$$

Then from (1) and (6) we obtain

$$\begin{aligned} J(K_0, K_1) &= E\{x_0^T \textstyle\sum_{t=0}^{\infty} ((\psi(K_0, K_1)^t)^T L(\cdot) \psi(K_0, K_1)^t) x_0\}, \\ &= E\{x_0^T U(K_0, K_1) x_0\} \\ &= Tr(U(K_0, K_1) V) \end{aligned} \tag{8}$$

where $V = E\{x_0^T x_0\}$ is the covariance matrix and

$$U(K_0, K_1) = \textstyle\sum_{t=0}^{\infty} (\psi(K_0, K_1)^t)^T L(K_0, K_1) \psi(K_0, K_1)^t$$

solves the discrete Lyapunov equation:

$$U(K_0, K_1) = \psi(K_0, K_1)^T U(K_0, K_1) \psi(K_0, K_1) + L(K_0, K_1). \tag{9}$$

By the trace properties it holds that; see Lemma 7.1

$$Tr(U(K_0, K_1) V) = Tr(L(K_0, K_1) P(K_0, K_1)), \tag{10}$$

where the matrix variable $P(K_0, K_1)$ solves the following discrete Lyapunov equation:

$$P(K_0, K_1) = \psi(K_0, K_1) P(K_0, K_1) \psi(K_0, K_1)^T + V.$$

Hence, the periodic discrete–time SOF problem (1) – (3) is stated as the following minimization problem:

$$\min J(K_0, K_1) = Tr(L(K_0, K_1) P(K_0, K_1)), \tag{11}$$

where $L(K_0, K_1)$ is as defined in (6) and the matrix variable $P(K_0, K_1)$ is the solution of the discrete Lyapunov equation:

$$P(K_0, K_1) = \psi(K_0, K_1) P(K_0, K_1) \psi(K_0, K_1)^T + V, \tag{12}$$

and K_0, K_1 must lie within the set $\mathcal{D}_d$ as defined in (5) .
The problem (11) – (12) is an unconstrained optimization problem in the matrix variables K_0, K_1, where the eigenvalue condition $K_0, K_1 \in \mathcal{D}_d$.
Note, that the set $\mathcal{D}_d$ is open and in general unbounded. Therefor, it is convenient to define the following level set:

$$\mathcal{L}(K_0, K_1) = \{K_0, K_1 \in \mathcal{D}_d : J(K_0, K_1) \leq J(K_0, K_1)\}. \tag{13}$$

This level set is compact; see [[19], Appendix A]. For given $K_0, K_1 \in \mathcal{D}_d$ the theorem of Bolzano– Weierstrass ensures the existence of a unique solution to the optimization problem (11) – (12) in the level set $\mathcal{L}(K_0, K_1)$; see [19].
The CG method was proposed by Hestenes and Stiefel [10] early in 1952 for solving linear systems of algebraic equations. Fletcher and Reeves [9] in 1964 developed a CG method for solving unconstrained optimization problems. Moreover, many different CG methods have been proposed in recent years (see, e.g,. [2, 7, 6] and the references therein). Recently, many literature suggested a hybridization of conjugate gradient methods for solving unconstrained optimization problems see [11, 12, 16]. The attempt in this paper is to apply HCG methods in papers [11, 12] for solving (11) – (12) . Moreover, the convergence theorems given in [11, 12] is extended to the considered algorithm.

2.1 Required derivatives of the objective function

The next Lemma provides a discrete Lyapunov equation required for obtaining the gradient of objective function (11) ; see [20].

Lemma 2.1 *Let K_0 and $K_1 \in \mathcal{D}_d$. Then $P(K_0,K_1)$ defined by* (12) *is differentiable and directional derivatives $\Delta P(\cdot)\Delta K_0$ and $\Delta P(\cdot)\Delta K_1$ of $P(K_0,K_1)$ are given by the discrete Lyapunov equations:* $\Delta P(\cdot)\Delta K_0 = \psi(\cdot)\Delta P(\cdot)\Delta K_0\psi(\cdot)^T + \psi(\cdot)P(\cdot)C_0^T\Delta K_0^T B_0^T\overline{A}_1^T + \overline{A}_1 B_0\Delta K_0 C_0 P(\cdot)\psi(\cdot)^T$ (14)

$$\Delta P(\cdot)\Delta K_1 = \psi(\cdot)\Delta P(\cdot)\Delta K_1\psi(\cdot)^T + \psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T B_1^T + B_1\Delta K_1 C_1\overline{A}_0 P(\cdot)\psi(\cdot)^T \quad (15)$$

where $\psi(\cdot) = \psi(K_0,K_1)$.

Proof: The directional derivatives of (12) with respect to K_0, K_1 respectively are given

$$\Delta P(\cdot)\Delta K_0 = \psi(\cdot)P(\cdot)C_0^T\Delta K_0^T B_0^T\overline{A}_1^T + [\psi(\cdot)\Delta P(\cdot)\Delta K_0 + \overline{A}_1 B_0\Delta K_0 C_0 P(\cdot)]\psi(\cdot)^T$$
$$= \psi(\cdot)\Delta P(\cdot)\Delta K_0\psi(\cdot)^T + \psi(\cdot)P(\cdot)C_0^T\Delta K_0^T B_0^T\overline{A}_1^T + \overline{A}_1 B_0\Delta K_0 C_0 P(\cdot)\psi(\cdot)^T,$$

$$\Delta P(\cdot)\Delta K_1 = \psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T B_1^T + [\psi(\cdot)\Delta P(\cdot)\Delta K_1 + B_1\Delta K_1 C_1\overline{A}_0 P(\cdot)]\psi(\cdot)^T$$
$$= \psi(\cdot)\Delta P(\cdot)\Delta K_1\psi(\cdot)^T + \psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T B_1^T + B_1\Delta K_1 C_1\overline{A}_0 P(\cdot)\psi(\cdot)^T. \ \square$$

The next lemma gives the first–order directional derivative of the objective function $J(K_0,K_1)$; see [20].

Lemma 2.2 *Consider the optimization problem* (11) - (12) *, where K_0 and $K_1 \in \mathcal{D}_d$. The first–order directional derivatives of the objective function* (11) *in the directions of ΔK_0 and ΔK_1 are given by*

$$J_{K_0}(\cdot)\Delta K_0 = 2Tr\left(\left(B_0^T(\overline{Q}_1\overline{A}_0 + \overline{A}_1 U(\cdot)\psi(\cdot) + R_0K_0C_0)\right)P(\cdot)C_0^T\Delta K_0^T\right), \quad (16)$$

$$J_{K_1}(\cdot)\Delta K_1 = 2Tr\left(\left(B_1^T U(\cdot)\psi(\cdot) + R_1K_1C_1\overline{A}_0\right)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T\right), \quad (17)$$

where $P(\cdot) = P(K_0,K_1)$ and $U(\cdot) = U(K_0,K_1)$ solve the discrete Lyapunov equations (12) and (9) respectively.

Proof: By differentiating the objective function with respect to K_0 in the direction of ΔK_0,

$$J_{K_0}(\cdot)\Delta K_0 = Tr(L(\cdot)\Delta P(\cdot)\Delta K_0) + Tr(\Delta L(\cdot)\Delta K_0 P(\cdot))$$
$$= Tr(L(\cdot)\Delta P(\cdot)\Delta K_0) + 2Tr\left((R_0K_0C_0 + B_0^T\overline{Q}_1\overline{A}_0)P(\cdot)C_0^T\Delta K_0^T\right).$$

From the Lyapunov equation (9) and (14) we have

$$Tr(L(\cdot)\Delta P(\cdot)\Delta K_0) = Tr\left(U(\cdot)\left(\psi(\cdot)P(\cdot)C_0^T\Delta K_0^T B_0^T\overline{A}_1^T\right)\right) + Tr\left(U(\cdot)\left(\overline{A}_1 B_0\Delta K_0 C_0 P(\cdot)\psi(\cdot)^T\right)\right)$$
$$= 2Tr\left(B_0^T\overline{A}_1^T U(\cdot)\psi(\cdot)P(\cdot)C_0^T\Delta K_0^T\right).$$

Hence, the directional derivative of the objective function in the direction of ΔK_0 is

$$J_{K_0}(\cdot)\Delta K_0 = 2Tr\left(\left(B_0^T\left(\overline{Q}_1\overline{A}_0 + \overline{A}_1^T U(\cdot)\psi(\cdot)\right) + R_0K_0C_0\right)P(\cdot)C_0^T\Delta K_0^T\right).$$

By differentiating the objective function with respect to K_1 in the direction of ΔK_1 we obtain,

$$J_{K_1}(\cdot)\Delta K_1 = Tr(L(\cdot)\Delta P(\cdot)\Delta K_1) + Tr(\Delta L(\cdot)\Delta K_1 P(\cdot))$$
$$= Tr(L(\cdot)\Delta P(\cdot)\Delta K_1) + 2Tr\left((R_1K_1C_1\overline{A}_0 P(\cdot)\overline{A}_0^T)C_1^T\Delta K_1^T\right).$$

From the discrete Lyapunov equation (9) and (15) we get;

$$Tr(L(\cdot)\Delta P(\cdot)\Delta K_1) = Tr\left(U(\cdot)\left(\psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T B_1^T\right)\right) + Tr\left(U(\cdot)\left(B_1\Delta K_1 C_1\overline{A}_0 P(\cdot)\psi(\cdot)^T\right)\right)$$
$$= 2Tr\left(B_1^T U(\cdot)\psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T\right),$$

we have,

$$J_{K_1}(\cdot)\Delta K_1 = 2Tr\left(\left(B_1^T U(\cdot)\psi(\cdot) + R_1K_1C_1\overline{A}_0\right)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T\right). \ \square$$

Let us write the directional derivative $J_{K_0}(\cdot)\Delta K_0$ as:

$$J_{K_0}(\cdot)\Delta K_0 = Tr(\nabla_{K_0}J(\cdot)\Delta K_0^T), \qquad \Delta K_0 \in \mathbb{R}^{p\times r}.$$

This implies that

$$\nabla_{K_0}J(\cdot) = 2\left(B_0^T(\overline{Q}_1\overline{A}_0 + \overline{A}_1 U(\cdot)\psi(\cdot) + R_0K_0C_0\right)P(\cdot)C_0^T). \quad (18)$$

Also

$$J_{K_1}(\cdot)\Delta K_1 = Tr(\nabla_{K_1}J(\cdot)\Delta K_1^T), \qquad \Delta K_1 \in \mathbb{R}^{p\times r}.$$

This implies that

$$\nabla_{K_1}J(\cdot) = 2\left(\left(B_1^T U(\cdot)\psi(\cdot) + R_1K_1C_1\overline{A}_0\right)P(\cdot)\overline{A}_0^T C_1^T\right). \quad (19)$$

The next Lemma yields the first-order necessary optimality condition for the optimization problem (11) and (12) .

Lemma 2.3 *Let K_0 and $K_1 \in \mathcal{D}_d$ be a local solution to the optimization problem* (11) *and* (12) *Then*

$$B_0^T\left(\overline{Q}_1\overline{A}_0 + \overline{A}_1 U(\cdot)\psi(\cdot) + R_0K_0C_0\right)P(\cdot)C_0^T = 0, \tag{20}$$

$$\left(B_1^T U(\cdot)\psi(\cdot) + R_1K_1C_1\overline{A}_0\right)P(\cdot)\overline{A}_0^T C_1^T = 0, \tag{21}$$

where P and U solve the discrete Lyapunov equation (12) and (9) , respectively.

III. Hybrid CG methods for the SOF design problem

In this section two hybrid CG methods are considered to tackle the optimization problem (11) and (12) . Moreover, these algorithms are extended to tackle the optimization problem originated from the SOF design problem for periodic discrete–time control systems. Global convergence is established for the proposed algorithm under a non–monotonic backtracking strategy.

Given $G_0 = [K_{0,0}^T \;\; K_{1,0}^T]^T \in \mathcal{D}_d$, the nonlinear hybrid CG methods HCG1 and HCG2 generate a sequence of iterates G_k according to the recurrence relation:

$$G_{k+1} = G_k + \alpha_k \Delta G_k \in \mathcal{D}_d, \quad k = 0,1,2,\ldots, \tag{22}$$

the search direction $\Delta G_k = [\Delta K_{0,k}^T \;\; \Delta K_{1,k}^T]^T$ is a descent direction for J at G_k that satisfies the following descent condition:

$$Tr(\nabla J(G_k)^T \Delta G_k) < 0, \quad k = 0,1,2,\cdots, \tag{23}$$

where $\nabla J(G_k) = [\nabla_{K_{0,k}} J(K_{0,k})^T \;\; \nabla_{K_{1,k}} J(K_{1,k})^T]^T$ is the gradient of the objective function J. Most of the HCG methods update the directions ΔG_k by the following relation:

$$\Delta G_{k+1} = -\nabla J(G_{k+1}) + \beta_k \Delta G_k, \quad \Delta G_0 = -\nabla J(G_0), \tag{24}$$

where β_k is a parameter that differs from one HCG method to the other.

The following Wolfe conditions, see e.g. [25], are used for updating a suitable step–size α_k for the calculated new iterate (22)

$$J(G_k + \alpha_k \Delta G_k) - J(G_k) \le \gamma \alpha_k Tr(\nabla J(G_k)^T \Delta G_k), \tag{25}$$

$$Tr(\nabla J(G_k + \alpha_k \Delta G_k)^T \Delta G_k) \ge \hat{\gamma} Tr(\nabla J(G_k)^T \Delta G_k), \tag{26}$$

where $0 < \gamma < \hat{\gamma} < 1$. Moreover, the strong Wolfe condition replaces (26) by the following condition

$$|Tr(\nabla J(G_k + \alpha_k \Delta G_k)^T \Delta G_k)| \le \hat{\gamma} |Tr(\nabla J(G_k)^T \Delta G_k)|. \tag{27}$$

3.1 The descent HCG1 method

Consider the unconstrained optimization problem (11) and (12) . The new search direction ΔG_{k+1} is obtained by using the following recurrence formula method [11]:

$$\Delta G_{k+1} = -\nabla J(G_{k+1}) + \beta_k^N \Delta G_k, \quad \Delta G_0 = -\nabla J(G_0), \tag{28}$$

where

$$\beta_k^N = \frac{\|\nabla J(G_{k+1})\|^2 - \max\left\{0, \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|} \cdot Tr(\nabla J(G_{k+1})^T \nabla J(G_k))\right\}}{\max\{\|\nabla J(G_k)\|^2, Tr(\Delta G_k^T Y_k)\}}, \tag{29}$$

and $S_k = G_{k+1} - G_k, Y_k = \nabla J(G_{k+1}) - \nabla J(G_k)$.

Note that β_k^N might be chosen as any of the following alternatives: β_k^{DY} or β_k^{FR} or β_k^{WYL} or β_k^{YWH} where

$$\beta_k^{DY} = \frac{\|\nabla J(G_{k+1})\|^2}{Tr(\Delta G_k^T Y_k)}, \qquad \beta_k^{FR} = \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2},$$

$$\beta_k^{WYL} = \frac{\|\nabla J(G_{k+1})\|^2 - \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|} \cdot Tr(\nabla J(G_{k+1})^T \nabla J(G_k))}{\|\nabla J(G_k)\|^2}$$

$$\beta_k^{YWH} = \frac{\|\nabla J(G_{k+1})\|^2 - \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|} \cdot Tr(\nabla J(G_{k+1})^T \nabla J(G_k))}{Tr(\Delta G_k^T Y_k)}.$$

Using the weak Wolfe line search rule (25) – (26) the HCG1 method (28) – (29) generates a descent direction to the objective function $J(G)$. This is proven in the next lemma. The parameter β_k^N is chosen such that the sufficient descent condition is satisfied every iteration, where the following conjugancy condition holds:

$$Tr(Y_k^T \Delta G_{k+1}) = 0.$$

Lemma 3.1 *If ΔG_{k+1} is evaluated by* (28) *and* (29) *such that $\Delta G_{k+1} \in \mathcal{D}_d$, then*

$$Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) < 0$$

holds for each $k \ge 0$.

Proof: (See also[11]) For $k = 0$, it is easy to have that

$$Tr(\nabla J(G_1)^T \Delta G_1) = -\|\nabla J(G_1)\|^2 < 0.$$

Assume that $Tr(\nabla J(G_k)^T \Delta G_k) < 0$ holds for k and $k > 1$. To get

$$Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) < 0,$$

for k we divide the proof into the following four cases. If $\beta_k^N = 0$, from (28) , one knows that

$$Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) = -\|\nabla J_{k+1}\|^2 < 0.$$

Therefor, in the analysis below, we always suppose $\beta_k^N \neq 0$.

Case(1): If $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) \leq 0$ and $Tr(\Delta G_k^T Y_k) \geq \|\nabla J(G_k)\|^2$, then from (29) we have

$$\beta_k^N = \frac{\|\nabla J(G_{k+1})\|^2}{Tr(\Delta G_k^T Y_k)} = \beta_k^{DY}.$$

Noting that $\|\nabla J(G_{k+1})\|^2 > 0$, so $Tr(\Delta G_k^T Y_k) > 0$ holds. Therefore from (29) , we have

$$\begin{aligned} Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) &= Tr(\nabla J(G_{k+1})^T(-\nabla J(G_k) + \beta_k^N \Delta G_k)) \\ &= -\|\nabla J(G_{k+1})\|^2 + \frac{\|\nabla J(G_{k+1})\|^2}{Tr(\Delta G_k^T Y_k)} \cdot Tr(\nabla J(G_{k+1})^T \Delta G_k) \\ &= \frac{\|\nabla J(G_{k+1})\|^2 \cdot Tr(\nabla J(G_k)^T \Delta G_k)}{Tr(\Delta G_k^T Y_k)} < 0. \qquad (30) \end{aligned}$$

Case(2): If $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) \leq 0$ and $Tr(\Delta G_k^T Y_k) < \|\nabla J(G_k)\|^2$, then from (29) one has

$$\beta_k^N = \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} = \beta_k^{FR}.$$

Therefore, in view of

$$Tr(\Delta G_k^T \nabla J(G_{k+1})) < \|\nabla J(G_k)\|^2 + Tr(\nabla J(G_k)^T \Delta G_k),$$

and (29) as well as (30) we obtain:

$$\begin{aligned} &Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) = Tr(\nabla J(G_{k+1})^T(-\nabla J(G_{k+1}) + \beta_k^N \Delta G_k) \\ &= -\|\nabla J(G_{k+1})\|^2 + \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} \cdot Tr(\nabla J(G_{k+1})^T \Delta G_k) \\ &< -\|\nabla J(G_{k+1})\|^2 \\ &+ \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2}(\|\nabla J(G_k)\|^2 + Tr(\nabla J(G_k)^T \Delta G_k)) \\ &= \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} \cdot Tr(\nabla J(G_k)^T \Delta G_k) < 0. \qquad (31) \end{aligned}$$

Case(3): If $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) > 0$ and $Tr(\Delta G_k^T Y_k) \geq \|\nabla J(G_k)\|^2$, then from (29) one has

$$\beta_k^N = \frac{Tr(\nabla J(G_{k+1})^T \nabla J(G_{k+1})) - \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|} Tr(\nabla J(G_{k+1})^T \nabla J(G_k))}{Tr(\Delta G_k^T Y_k)} = \beta_k^{YWH}.$$

Noticing that $\beta_k^N \neq 0$ and $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) > 0$, we have $0 < cos\theta_k < 1$, where θ_k is the angle between $\nabla J(G_{k+1})$ and $\nabla J(G_k)$. Thus from (28) – (29) one has

$$\begin{aligned} &Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) = Tr(\nabla J(G_{k+1})^T(-\nabla J(G_{k+1}) + \beta_k^N \Delta G_k)) = -\|\nabla J(G_{k+1})\|^2 \\ &+ \frac{Tr(\nabla J(G_{k+1})^T(\nabla J(G_{k+1}) - \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|}\nabla J(G_k)))}{Tr(\Delta G_k^T Y_k)} \cdot Tr(\nabla J(G_{k+1})^T \Delta G_k) \\ &= -\|\nabla J(G_{k+1})\|^2 \\ &+ \frac{\|\nabla J(G_{k+1})\|^2 Tr(\nabla J(G_{k+1})^T \Delta G_k) - \|\nabla J(G_{k+1})\|^2 Tr(\nabla J(G_{k+1}) \Delta G_k) \cos\theta_k}{Tr(\Delta G_k^T Y_k)} \\ &= \frac{\|\nabla J(G_{k+1})\|^2 Tr(\nabla J(G_k)^T \Delta G_k) - \|\nabla J(G_{k+1})\|^2 \cos\theta_k Tr(\nabla J(G_{k+1})^T \Delta G_k)}{Tr(\Delta G_k^T Y_k)} \\ &< \frac{\|\nabla J(G_{k+1})\|^2 Tr(\nabla J(G_k)^T \Delta G_k) - \|\nabla J(G_{k+1})\|^2 \cos\theta_k Tr(\nabla J(G_k)^T \Delta G_k)}{Tr(\Delta G_k^T Y_k)} \\ &= \frac{\|\nabla J(G_{k+1})\|^2(1 - \cos\theta_k)}{Tr(\Delta G_k^T Y_k)} Tr(\nabla J(G_k)^T \Delta G_k) < 0. \qquad (32) \end{aligned}$$

Case(4): If $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) > 0$ and $Tr(\Delta G_k^T Y_k) < \|\nabla J(G_k)\|^2$, then from (29) it holds that

$$\beta_k^N = \frac{Tr(\nabla J(G_{k+1})^T \nabla J(G_{k+1})) - \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|} Tr(\nabla J(G_{k+1})^T \nabla J(G_k))}{\|\nabla J(G_k)\|^2} = \beta_k^{WYL}.$$

Similar to the analysis of the third case we have

$$\begin{aligned} &Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) = Tr(\nabla J(G_{k+1})^T(-\nabla J(G_{k+1}) + \beta_k^N \Delta G_k)) \\ &= -\|\nabla J(G_{k+1})\|^2 + \frac{Tr(\nabla J(G_{k+1})^T(\nabla J(G_{k+1}) - \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|}\nabla J(G_k)))}{\|\nabla J(G_k)\|^2} \\ &\times Tr(\nabla J(G_{k+1})^T \Delta G_k) \\ &= -\|\nabla J(G_{k+1})\|^2 + \frac{\|\nabla J(G_{k+1})\|^2(1 - \cos\theta_k)}{\|\nabla J(G_k)\|^2} Tr(\nabla J(G_{k+1})^T \Delta G_k) \\ &< -\|\nabla J(G_{k+1})\|^2 \\ &+ \frac{\|\nabla J(G_{k+1})\|^2(1 - \cos\theta_k)}{\|\nabla J(G_k)\|^2}(\|\nabla J(G_k)\|^2 + Tr(\nabla J(G_k)^T \Delta G_k)) \\ &= -\|\nabla J(G_{k+1})\|^2 \cos\theta_k + \frac{\|\nabla J(G_{k+1})\|^2(1 - \cos\theta_k)}{\|\nabla J(G_k)\|^2} Tr(\nabla J(G_k)^T \Delta G_k) \end{aligned}$$

$$< \frac{\|\nabla J(G_{k+1})\|^2(1-\cos\theta_k)}{\|\nabla J(G_k)\|^2} Tr(\nabla J(G_k)^T \Delta G_k) < 0. \quad (33)$$

Therefore, for all $k \geq 0$, $Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) < 0$ always holds. □

From the proof of Lemma 3.1 we can easily obtain the following important property about the formula (29) .

Lemma 3.2 *For any* $k \geq 0$*, the relation*

$$0 \leq \beta_k^N \leq \frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)}$$

always holds.

Proof: (See also [11]) From equation (29) , we have $\beta_k^N \geq 0$. If $\beta_k^N = 0$ and $\nabla J(G_{k+1}) \neq 0$ by Lemma 3.1 we have

$$\frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)} > 0 = \beta_k^N.$$

Assume that $\beta_k^N > 0$, we now prove

$$\beta_k^N \leq \frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)}$$

by considering the following four cases:

Case(1): If $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) \leq 0$ and $Tr(\Delta G_k^T Y_k) \geq \|\nabla J(G_k)\|^2$, then $\beta_k^N = \beta_k^{DY}$. Furthermore, from Lemma 3.1 and the formula (30) we have

$$\beta_k^N = \frac{\|\nabla J(G_{k+1})\|^2}{Tr(\Delta G_k^T Y_k)} = \frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)}.$$

Case(2): If $Tr(\nabla J(G_{k+1})^T \Delta G_k) \leq 0$ and $Tr(\Delta G_k^T Y_k) < \|\nabla J(G_k)\|^2$, then $\beta_k^N = \beta_k^{FR}$. Furthermore, form Lemma 3.1 and formula (31) , we have

$$\beta_k^N = \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} < \frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)}.$$

Case(3): If $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) > 0$ and $Tr(\Delta G_k^T Y_k) \geq \|\nabla J(G_k)\|^2$, then β_k^N reduces to β_k^{YWH}. Furthermore, by Lemma 3.1 and formula (32) one has

$$\beta_k^N = \frac{Tr(\nabla J(G_{k+1})^T \nabla J(G_{k+1})) - \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|} Tr(\nabla J(G_{k+1})^T \nabla J(G_k))}{Tr(\Delta G_k^T Y_k)}$$
$$= \frac{\|\nabla J(G_k)\|^2(1-\cos\theta_k)}{Tr(\Delta G_k^T Y_k)} < \frac{Tr(\nabla J(G_{k+1}) \Delta G_{k+1})}{Tr(\nabla J(G_k) \Delta G_k)}.$$

Case(4): If $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) > 0$ and $Tr(\Delta G_k^T Y_k) < \|\nabla J(G_k)\|^2$, then $\beta_k^N = \beta_k^{WYL}$. Furthermore, by Lemma 3.1 and formula (33) it follow that

$$\beta_k^N = \frac{Tr(\nabla J(G_{k+1})^T \nabla J(G_{k+1})) - \frac{\|\nabla J(G_{k+1})\|}{\|\nabla J(G_k)\|} Tr(\nabla J(G_{k+1})^T \nabla J(G_k))}{\|\nabla J(G_k)\|^2}$$
$$= \frac{\|\nabla J(G_k)\|^2(1-\cos\theta_k)}{\|\nabla J(G_k)\|^2} < \frac{Tr(\nabla J(G_{k+1}) \Delta G_{k+1})}{Tr(\nabla J(G_k) \Delta G_k)}.$$

Therefore Lemma 3.2 holds. □

3.2 The descent HCG2 method

In this section we consider the second alternative of hybrid CG methods HCG2 (see also [12]). It combines the Fletcher–Reeves and Polak–Ribière–Polyak CG methods.

The new search direction ΔG_{k+1} is generated by (24) , where the updating parameter β_k is given by

$$\beta_k^{PR-FR} = (1 - \hat{\theta}_k)\beta_k^{PRP+} + \hat{\theta}_k \beta_k^{FR}, \quad (34)$$
$$\beta_k^{PRP+} = \max\left\{\frac{Tr(\nabla J(G_{k+1})^T Y_k)}{\|\nabla J(G_k)\|^2}, 0\right\}, \quad \beta_k^{FR} = \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2}$$

The hybridization parameter $\hat{\theta}_k$ is computed by

$$\hat{\theta}_k = \begin{cases} \theta_k^*, & if \quad Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) \neq 0 \quad and \quad \theta_k^* \in [0,1], \\ 0, & if \quad \theta_k^* < 0, \quad or \quad Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) = 0, \\ 1, & if \quad \theta_k^* > 1. \end{cases} \quad (35)$$

where

$$\theta_k^* = -\frac{Tr(\nabla J(G_{k+1})^T \Delta G_k)}{Tr(\nabla J(G_{k+1})^T \nabla J(G_k))} \frac{Tr(\Delta G_k^T Y_k)}{\|\Delta G_k\|^2}.$$

Lemma 3.3 *If* ΔG_{k+1} *is evaluated by* (24) *and* (34) *such that* $\Delta G_{k+1} \in \mathcal{D}_d$*, then*

$$Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) < 0$$

holds for each $k \geq 0$.

Proof: For $k=0$, it is easy to have that

$$Tr(\nabla J(G_1)^T \Delta G_1) = -\|\nabla J(G_1)\|^2 < 0.$$

Assume that $Tr(\nabla J(G_k)^T \Delta G_k) < 0$ holds for k and $k \geq 1$. To get

$$Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) < 0,$$

for k we divide the proof into the following two cases. If $\beta_k^{PR-FR} = 0$, from (24) , one knows that

$$Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) = -\|\nabla J_{k+1}\|^2 < 0.$$

Therefor, in the analysis below, we always suppose $\beta_k^{PR-FR} \neq 0$.

Case(1): If $Tr(\nabla J(G_{k+1})^T Y_k) \leq 0$ and $\theta_k^* > 1$ then $\hat{\theta}_k = 1$, from (34) has one

$$\beta_k^{PR-FR} = \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} = \beta_k^{FR}.$$

Therefore, in view of

$$Tr(\Delta G_k^T \nabla J(G_{k+1})) < \|\nabla J(G_k)\|^2 + Tr(\nabla J(G_k)^T \Delta G_k),$$

and (24) as well as (34) we obtain:

$$\begin{aligned}
&Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) = Tr(\nabla J(G_{k+1})^T(-\nabla J(G_{k+1}) + \beta_k^{PR-FR} \Delta G_k) \\
&= -\|\nabla J(G_{k+1})\|^2 + \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} \cdot Tr(\nabla J(G_{k+1})^T \Delta G_k) \\
&< -\|\nabla J(G_{k+1})\|^2 \\
&+ \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2}(\|\nabla J(G_k)\|^2 + Tr(\nabla J(G_k)^T \Delta G_k)) \\
&= \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} \cdot Tr(\nabla J(G_k)^T \Delta G_k) < 0.
\end{aligned} \tag{36}$$

Case(2): If $Tr(\nabla J(G_{k+1})^T Y_k) > 0$ and $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) = 0$ or $\theta_k^* < 0$ then $\hat{\theta}_k = 0$, from (34) has one

$$\beta_k^{PR-FR} = \frac{Tr(\nabla J(G_{k+1})^T Y_k)}{\|\nabla J(G_k)\|^2} = \beta_k^{PR},$$

from (24) as well as (34) we obtain:

$$\begin{aligned}
&Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) = Tr(\nabla J(G_{k+1})^T(-\nabla J(G_{k+1}) + \beta_k^{PR-FR} \Delta G_k) \\
&= -\|\nabla J(G_{k+1})\|^2 + \frac{Tr(\nabla J(G_{k+1})^T Y_k)}{\|\nabla J(G_k)\|^2} \cdot Tr(\nabla J(G_{k+1})^T \Delta G_k) \\
&< -\|\nabla J(G_{k+1})\|^2 \\
&+ \frac{Tr(\nabla J(G_{k+1})^T Y_k)}{\|\nabla J(G_k)\|^2}(\|\nabla J(G_k)\|^2 + Tr(\nabla J(G_k)^T \Delta G_k)) \\
&< -\|\nabla J(G_{k+1})\|^2 + Tr(\nabla J(G_{k+1})^T Y_k) \\
&+ \frac{Tr(\nabla J(G_{k+1})^T Y_k)}{\|\nabla J(G_k)\|^2} Tr(\nabla J(G_k)^T \Delta G_k) \\
&= -Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) \\
&+ \frac{Tr(\nabla J(G_{k+1})^T Y_k)}{\|\nabla J(G_k)\|^2} Tr(\nabla J(G_k)^T \Delta G_k) \\
&\leq \frac{Tr(\nabla J(G_{k+1})^T Y_k)}{\|\nabla J(G_k)\|^2} \cdot Tr(\nabla J(G_k)^T \Delta G_k) < 0.
\end{aligned} \tag{37}$$

Therefor, for all $k \geq 0$, $Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) < 0$ always holds. □

From the proof of Lemma 3.3 we can easily obtain the following important property about the formula (34) .

Lemma 3.4 *For any $k \geq 0$, the relation*

$$0 \leq \beta_k^{PR-FR} \leq \frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)}$$

always holds.

Proof: From equation (34) , we have $\beta_k^{PR-FR} \geq 0$. If $\beta_k^{PR-FR} = 0$ and $\nabla J(G_{k+1}) \neq 0$ by Lemma 3.3 we have

$$\frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)} > 0 = \beta_k^{PR-FR}.$$

Assume that $\beta_k^{PR-FR} > 0$, we now prove

$$\beta_k^{PR-FR} \leq \frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)},$$

by considering the following two cases:

Case(1): If $Tr(\nabla J(G_{k+1})^T Y_k) \leq 0$ and $\theta_k^* > 1$ then $\hat{\theta}_k = 1$, $\beta_k^{PR-FR} = \beta_k^{FR}$.

Furthermore, from Lemma 3.3 and the formula (36) we have

$$\beta_k^{PR-FR} = \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} < \frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)}.$$

Case(2): If $Tr(\nabla J(G_{k+1})^T Y_k) > 0$ and $Tr(\nabla J(G_{k+1})^T \nabla J(G_k)) = 0$ or $\theta_k^* < 0$ then $\hat{\theta}_k = 0$ and $\beta_k^{PR-FR} = \beta_k^{PR}$. Furthermore, form Lemma 3.3 and formula (37) we have

$$\beta_k^{PR-FR} = \frac{Tr(\nabla J(G_{k+1})^T Y_k)}{\|\nabla J(G_k)\|^2} < \frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)}.$$

Therefore Lemma 3.4 holds. ▯

The overall algorithm for solving the minimization problem (11) – (12) is stated in the following lines, where any of the considered two updates of β_k can be used.

Algorithm 3.1 (Nonlinear HCG methods for solving Problems (11) – (12))

1. Initialization: Let $G_0 = [K_{0,0}^T \; K_{1,0}^T]^T$ where $K_{0,0}, K_{1,0} \in \mathcal{D}_d$ be given and let $\epsilon^{tol} \in (0,1)$ be the given tolerance. Choose $0 < \gamma < \hat{\gamma} < 1$ and calculate $P(K_{0,0}, K_{1,0})$ and $U(K_{0,0}, K_{1,0})$ solution of Lyapunov equations (12) and (9) , respectively. Calculate $\|\nabla J(G_0)\|$; Set $\Delta G_0 := -\nabla J(G_0)$ and $k \leftarrow 0$.

2. **While** $\|\nabla J(G_k)\| \geq \epsilon^{tol}$, do

(a) Calculate the first element α_k of a decreasing sequence, e.g.,$\{1/2^j\}_{j\geq 0}$ that satisfies the weak Wolf conditions (25) – (26) and the stability condition (22) , i.e., $G_k + \alpha_k \Delta G_k \in \mathcal{D}_d$.

(b) Set $G_{k+1} := G_k + \alpha_k \Delta G_k$

(c) Given $G_{k+1} = [K_{0,k+1}^T \; K_{1,k+1}^T]^T$ solve the discrete Lyapunov equations (12) and (9) for $P(K_{0,k+1}, K_{1,k+1})$ and $U(K_{0,k+1}, K_{1,k+1})$, respectively. Then calculate $\nabla J(G_{k+1})$; Set $Y_k := \nabla J(G_{k+1}) - \nabla J(G_k)$ and $S_k := G_{k+1} - G_k$; Calculate β_k by one of the formulas (29) or (34)

(d) Calculate:

$$\Delta G = -\nabla J(G_{k+1}) + \beta_k \Delta G_k$$

and choose the new direction as:

$$\Delta G_{k+1} = \begin{cases} \Delta G, & Tr(\nabla J(G_{k+1})^T \Delta G) \leq -10^{-3} \|\Delta G\| \|\nabla J(G_{k+1})\| \\ -\nabla J(G_{k+1}), & otherwise. \end{cases} \tag{38}$$

(e) Set $k \leftarrow k + 1$ and go to (a).

3. **End** (do)

IV. Convergence analysis

The convergence analysis of Algorithm 3.1 is established in this section. Next, assume that $\nabla J(G_k) \neq 0$ for all k; otherwise a stationary point is found.

Assumption 4.1 *The following assumptions hold:*

1. The objective function J is bounded from below.
2. The level set (13) is bounded.
3. In some neighborhood $\bar{\mathcal{N}}$ of $\bar{L}$, J is continuously differentiable, and its gradient is Lipschitz continuous, namely, there exists a constant $(\bar{L} > 0)$ such that

$$\|\nabla J(G_1) - \nabla J(G_2)\| \leq \bar{L} \|G_1 - G_2\| \quad \forall \; G_1, G_2 \in \bar{\mathcal{N}} \tag{39}$$

4. At any iteration k there always exists an $\alpha_k > 0$ such that $G_k + \alpha_k \Delta G_k \in \mathcal{D}_d$.

The following Lemmas are used for proving the main global convergence theorem, see [5] for the proofs.

Lemma 4.1 *Consider the optimization problem* (11) – (12) *, let* $\{G_k\} \subset \mathcal{D}_d$ *be generated by Algorithm* 3.1 *and assume that Assumption* 4.1 *holds. then the Wolfe conditions* (25) – (26) *is feasible.*

Lemma 4.2 *Let* ΔG_k *is given by* (28) – (29) *. Then we have*

$$Tr(\nabla J(G_k)^T \Delta G_k) \leq -\frac{7}{8} \|\nabla J(G_k)\|^2,$$

holds for any $k \geq 0$, i.e, ΔG_k is descent direction for $J(G)$.

Lemma 4.3 *Let* $G \in \mathcal{D}_d$ *be generated by Algorithm* 3.1 *and assume that* ΔG_k *is a descent direction. Furthermore, let Assumption* 4.1 *holds. Then*

$$\sum_{k=1}^{\infty} \frac{(Tr(\nabla J(G_k)^T) \Delta G_k)^2}{\|\Delta G_k\|^2} < +\infty.$$

Proof: (See also [5]) From (25) , (26) , Lemma 4.2 and Assumption 4.1 we obtain

$$(2\hat{\gamma} + \bar{L})\alpha_k \|\Delta G_k\|^2 \geq -Tr(\nabla J(G_k^T \Delta G_k)).$$

Then we have

$$\alpha_k \|\Delta G_k\| \geq \frac{1}{2\hat{\gamma} + \bar{L}} \left(-\frac{Tr(\nabla J(G_k)^T \Delta G_k)}{\|\Delta G_k\|} \right).$$

Squaring both sides of the above formula we obtain

$$\alpha_k^2 \|\Delta G_k\|^2 \geq \left(\frac{1}{2\hat{\gamma} + \bar{L}} \right)^2 \left(\frac{Tr(\nabla J(G_k)^T \Delta G_k)}{\|\Delta G_k\|} \right)^2.$$

From (25) we have

$$\sum_{k=1}^{\infty} \frac{(Tr(\nabla J(G_k)^T \Delta G_k))^2}{\|\Delta G_k\|^2} \leq (2\hat{\gamma} + \bar{L})^2 \sum_{k=1}^{\infty} \alpha_k^2 \|\Delta G_k\|^2 \frac{(2\hat{\gamma} + L)^2}{\gamma} \times \sum_{k=1}^{\infty} |J(G_k) - J(G_{k+1})| < +\infty.$$

4.1 Convergence result for the method HCG1

The following theorem presents the global convergence result of Algorithm 3.1 , see [11, Theorem 6].

Theorem 4.1 *Let Assumption* 4.1 *holds and* $G_k \in \mathcal{D}_d$ *be generated by Algorithm* 3.1 . *Then*

$$\lim_{k\to\infty} \inf \| \nabla J(G_k) \| = 0. \tag{40}$$

Proof: Suppose by contradiction that the stated conclusion is not true. Then in view of $\|\nabla J(G_k)\| > 0$ there exists a constant $\gamma > 0$ such that $\|\nabla J(G_k)\|^2 \geq 0$.

From (28) it follows that $\Delta G_{k+1} + \nabla J(G_k) = \beta_k^N \Delta G_k$. This together with Lemma 3.2 implies

$$\begin{aligned}\|\Delta G_{k+1}\|^2 &= (\beta_k^N)^2 \|\Delta G_k\|^2 - 2Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) - \|\nabla J(K_{k+1})\|^2 \\ &\leq \left(\frac{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}{Tr(\nabla J(G_k)^T \Delta G_k)}\right)^2 \|\Delta G_k\|^2 \\ &-2Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}) - \|\nabla J(G_{k+1})\|^2. \end{aligned} \tag{41}$$

Dividing the both sides of (41) by $(Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}))^2$ we obtain

$$\begin{aligned}\frac{\|\Delta G_{k+1}\|^2}{(Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}))^2} &\leq \frac{\|\Delta G_k\|^2}{(Tr(\nabla J(G_k)^T \Delta G_k))^2} \\ &- \frac{2}{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})} - \frac{\|\nabla J(G_{k+1})\|^2}{(Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}))^2} \\ &= \frac{\|\Delta G_k\|^2}{(Tr(\nabla J(G_k)^T \Delta G_k))^2} \\ &- \left(\frac{1}{\|\nabla J(G_{k+1})\|} + \frac{\|\nabla J(G_{k+1})\|}{Tr(\nabla J(G_{k+1})^T \Delta G_{k+1})}\right)^2 \\ &+ \frac{1}{\|\nabla J(G_{k+1})\|^2} \\ &\leq \frac{\|\Delta G_k)\|^2}{(Tr(\nabla J(G_k)^T \Delta G_k))^2} + \frac{1}{\|\nabla J(G_{k+1})\|^2}. \end{aligned} \tag{42}$$

Combining with

$$\frac{\|\Delta G_1\|^2}{(Tr(\nabla J(G_1)^T \Delta G_1))^2} = \frac{1}{\|\nabla J(G_1)\|^2},$$

by the recurrence relation (42) and $\|\nabla J(G_{k+1})\|^2 \geq \gamma$ we have

$$\begin{aligned}\frac{\|\Delta G_{k+1}\|^2}{(Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}))^2} &\leq \frac{\|\Delta G_k\|^2}{(Tr(\nabla J(G_k)^T \Delta G_k))^2} + \frac{1}{\|\nabla J(G_{k+1})\|^2} \\ &\leq \frac{\|\Delta G_{k-1}\|^2}{(Tr(\nabla J(G_{k-1})^T \Delta G_{k-1}))^2} + \frac{1}{\|\nabla J(G_k)\|^2} + \frac{1}{\|\nabla J(G_{k+1})\|^2} \\ &\leq \cdots \leq \sum_{i=1}^{k} \frac{1}{\|\nabla J(G_i)\|^2} \leq \frac{k}{\gamma}. \end{aligned} \tag{43}$$

Thus,

$$\frac{(Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}))^2}{\|\Delta G_{k+1}\|^2} \geq \frac{\gamma}{k}.$$

This further shows that

$$\sum_{k=1}^{\infty} \frac{(Tr(\nabla J(G_{k+1})^T \Delta G_{k+1}))^2}{\|\Delta G_{k+1}\|^2} = \infty,$$

which contradicts Lemma 4.3. Therefore, the desired result holds. □

4.2 Convergence result for the method HCG2

The convergence result for the HCG2 method under Wolfe condition (25) and the follow weakened version of (26) is given

$$Tr(\nabla J(G_k + \alpha_k \Delta G_k)^T \Delta G_k) \geq \hat{\gamma} Tr(\nabla J(G_k)^T \Delta G_k), \tag{44}$$

where $0 < \gamma < \hat{\gamma}$.

We need the following theorem which is established by Zoutendijk [30] under Wolfe conditions.

Theorem 4.2 *Assume that* $G \in \mathcal{D}_d$ *the Assumption* 4.1 *holds and* $\Delta G_k, k \geq 0$ *is a descent direction where the step length* α_k *satisfies Wolfe conditions* (25) – (44) . *Then*

$$\sum_{k=0}^{\infty} \cos^2\phi_k \|\nabla J(G_k)\|^2 < \infty, \tag{45}$$

where

$$\cos\phi_k = -\frac{Tr(\nabla J(G_k)^T \Delta G_k)}{\|\nabla J(G_k)\| \|\Delta G_k\|}.$$

Theorem 4.3 *Suppose that Assumption* 4.1 *holds, the step* $\Delta G_k, k \geq 0$ *is evaluated by* (28) *and* (34) *with the following three properties*

1. $\beta_k^{PR-FR} \geq 0, \quad \forall \quad k \geq 0.$
2. The line search satisfies $\{G_k\}_{k\geq 0} \subset \mathcal{L}$ and the sufficient descent condition

$$Tr(\nabla J(G_k)^T \Delta G_k) \leq c\|\nabla J(G_k)\|^2, \quad \forall \ k \geq 0, \quad c > 0$$

and the Zoutendijk condition (45)

3. Suppose that the following inequality holds

$$0 < \hat{\gamma} \leq \|\nabla J(G_k)\| < \gamma, \quad \forall \ k \geq 0. \tag{46}$$

if there exist constants $b_1 > 1$ and $b_2 > 0$ such that $\forall \ k$,

$$|\beta_k^{PR-FR}| \leq b_1 \tag{47}$$

and

$$\|S_k\| \leq b_2 \Rightarrow |\beta_k^{PR-FR}| \leq \frac{1}{2b_1}. \tag{48}$$

Then

$$\lim_{k\to\infty} \ \inf\|\nabla J(G_k)\| = 0. \tag{49}$$

Theorem 4.4 *Suppose that Assumption* 4.1 *holds and the sequence* $\{G_k\}$ *be generated by Algorithm* 3.1 *and there exist a positive constant* $\widetilde{M}$ *such that*

$$\hat{\theta}_k \leq \widetilde{M}\|S_k\|. \tag{50}$$

Then (49) holds.

Proof: By the Wolfe condition (25) the sequence $\{G_k\}_{k\geq 0}$ is subset of the level set $\mathcal{L}$ also from Theorem 4.2 , the Zoutendijk condition holds. Therefore, considering Theorem 4.3 , to complete the proof it is enough to show that the formula (46) holds. Since $\hat{\theta}_k \in [0,1]$, from (46) we have

$$\begin{aligned} |\beta_k^{PR-FR}| &\leq \beta_k^{PRP+} + \beta_k^{FR} \\ &\leq \frac{|Tr(\nabla J(G_{k+1})^T Y_k)|}{\|\nabla J(G_k)\|^2} + \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} \\ &\leq \frac{\|\nabla J(G_{k+1})\|\|Y_k\|}{\|\nabla J(G_k)\|^2} + \frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} \\ &\leq \frac{\gamma\times 2\gamma}{\hat{\gamma}^2} + \frac{\gamma^2}{\hat{\gamma}^2} = 3\frac{\gamma^2}{\hat{\gamma}^2} \end{aligned} \tag{51}$$

from (39) and (50)

$$\begin{aligned} |\beta_k^{PR-FR}| &\leq \beta_k^{PRP+} + \hat{\theta}_k\beta_k^{FR} \\ &\leq \frac{|Tr(\nabla J(G_{k+1})^T Y_k)|}{\|\nabla J(G_k)\|^2} + \widehat{M}\|S_k\|\frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} \\ &\leq \frac{\|\nabla J(G_{k+1})\|\|Y_k\|}{\|\nabla J(G_k)\|^2} + \widehat{M}\|S_k\|\frac{\|\nabla J(G_{k+1})\|^2}{\|\nabla J(G_k)\|^2} \\ &\leq \frac{\bar{L}\gamma\|S_k\|}{\hat{\gamma}^2} + \widehat{M}\|S_k\|\frac{\gamma^2}{\hat{\gamma}^2} \\ &\leq \frac{\bar{L}\gamma + \widehat{M}\gamma^2}{\hat{\gamma}^2}\|S_k\| \end{aligned} \tag{52}$$

Therefore, from (51) and (52) if we let

$$b_1 = \frac{3\gamma^2}{\hat{\gamma}^2}, \ and \quad b_2 = \frac{\hat{\gamma}^2}{2b_1(\bar{L}\gamma + \widehat{M}\gamma^2)},$$

then (47) and (48) hold, Thus the formula (46) holds.

5 Numerical results of the SOF problem

In this section an implementation of the two algorithms HCG1 and HCG2 are described. Two MATLAB codes were written corresponding to this implementation. The two methods are compared numerically with the classical PRP conjugate gradient method (see e.g. [25]).

We compared performance of the HCG1, HCG2 and PRP methods with respect to number of iterations, CPU time and number of wins. In table 1 the first to the fourth columns are, respectively, the iteration counter k, the objective function $J(K_0, K_1, K_2)$, the convergence criterion $\|\nabla J(K_0, K_1, K_2)\|$ and the spectral radius $\rho(A(K_0, K_1, K_2))$ as indicator of fulfilling the stability condition. For every iteration of the HCG1, HCG2 and PRP methods two discrete Lyapunov equations are solved using the MATLAB function $dlyap(\cdot,\cdot)$. The following values have been assigned to the parameters of Algorithm 3.1 :

$$\gamma = 10^{-4}, \hat{\gamma} = 0.1, \epsilon^{tol} = 10^{-4},$$

Instead of using the Wolfe conditions (25) – (26) we have also tried the simple sufficient decrease condition (25) with $\gamma = 10^{-4}$, where the initial step size is chosen as $\alpha_0 = 1$ in the backtracking rule. The methods have given satisfactory results using this alternative.

The following examples quite show the performance of the HCG methods for finding a local solution to the problem (8) – (12) . A starting feasible point for the method might be obtained by executing the PSO method, see

[24]. In the following four examples are given for testing the proposed method on periodic systems, where in the first two examples the period is taken as $d = 2$ and in the second two $d = 3$.

Example 5.1 This test problem is borrowed from [15]. The discrete–time counterpart has the following data matrices **with period $d = 2$**:

$$A_0 = \begin{bmatrix} 0.9765 & -0.0265 & 0 \\ -0.4151 & 0.9314 & 0 \\ -0.0211 & 0.0964 & 1.0000 \end{bmatrix}, A_1 = \begin{bmatrix} 0.9628 & -0.0316 & 0 \\ -0.2740 & 0.9084 & 0 \\ -0.0140 & 0.0953 & 1.0000 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 0.0125 \\ -0.0770 \\ -0.0039 \end{bmatrix}, B_1 = \begin{bmatrix} 0.0163 \\ -0.0984 \\ -0.0050 \end{bmatrix}, C_0 = C_1 = I_3,$$

$$Q_0 = Q_1 = I_3, \quad R_0 = R_1 = I_1, \quad V = I_3.$$

The uncontrolled system is not discrete–time Schur stable, where $\rho(A_1A_0) = 1.0980$. Starting with the following initial $K_{0,0}, K_{1,0} \in \mathcal{D}_d$ the methods HCG1, HCG2 and PRP require 27, 55 and 37 iterations with CPU times 0.35, 0.62 and 0.51, respectively, to reach the stationary point $K_{0,fin}, K_{1,fin}$. The starting and final feedback gain matrices are:

$$K_{0,0} = \begin{bmatrix} 0.6806 & -0.5981 & 0.1704 \end{bmatrix}, \quad K_{1,0} = \begin{bmatrix} -3.3851 & 15.2394 & 2.7762 \end{bmatrix},$$

$$K_{0,fin} = \begin{bmatrix} -3.0357 & 1.2399 & 0.8052 \end{bmatrix}, \quad K_{1,fin} = \begin{bmatrix} -3.6371 & 1.5328 & 1.0008 \end{bmatrix}.$$

Example 5.2 This test problem is borrowed from [14]. The discrete–time counterpart has the following data matrices **with period $d = 2$**:

$$A_0 = A_1 = \begin{bmatrix} 1.0050 & 0.1002 \\ 0.1002 & 1.0050 \end{bmatrix}, B_0 = B_1 = \begin{bmatrix} 0.0050 \\ 0.1002 \end{bmatrix}, C_0^T = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_1^T = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$Q_0 = Q_1 = I_2, \quad R_0 = R_1 = I_1, \quad V = I_2.$$

The uncontrolled system is not discrete–time Schur stable, where $\rho(A_1A_0) = 1.2214$. Starting with the following initial $K_{0,0}, K_{1,0} \in \mathcal{D}_d$ the methods HCG1, HCG2 and PRP require 17, 13 and 18 iterations with CPU times 0.29, 0.23 and 0.29, respectively, to reach the stationary point K_{fin0}, K_{fin1}. The starting and final feedback gain matrices are:

$$K_{0,0} = \begin{bmatrix} -2.3425 \end{bmatrix}, \quad K_{1,0} = \begin{bmatrix} -0.6390 \end{bmatrix},$$

$$K_{0,fin} = [-3.4398], \quad K_{1,fin} = \begin{bmatrix} -2.1348 \end{bmatrix}.$$

Example 5.3 This test problem is borrowed from [28] having the following data matrices **with period $d = 3$**:

$$A_0 = \begin{bmatrix} 1.0000 & 0.1000 \\ 0 & 1.0000 \end{bmatrix}, A_1 = \begin{bmatrix} 1.1052 & 0.2103 \\ 0 & 1.0000 \end{bmatrix}, A_2 = \begin{bmatrix} 1.0000 & 0 \\ 0.1230 & 1.4918 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 0.3 \\ 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0.0103 \\ 0.1000 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 0.1230 \end{bmatrix}, C_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, C_2 = \begin{bmatrix} 3 \\ 1 \end{bmatrix},$$

$$Q_0 = Q_1 = Q_2 = I_3, \quad R_0 = R_1 = R_2 = I_1, \quad V = I_3.$$

The uncontrolled system is not discrete–time Schur stable, where $\rho(A_2A_1A_0) = 1.6165$. Starting with the following initial $K_{0,0}, K_{1,0}, K_{2,0} \in \mathcal{D}_d$ the methods HCG1, PRP require 24 and 23 iterations with CPU times 0.34 and 0.34, respectively, while the method HCG2 failed to converge the stationary point $K_{0,fin}, K_{1,fin}, K_{2,fin}$. The starting and final feedback gain matrices are:

$$K_{0,0} = \begin{bmatrix} -0.2297 \end{bmatrix}, \quad K_{1,0} = \begin{bmatrix} -1.0370 \end{bmatrix}, \quad K_{2,0} = \begin{bmatrix} -0.5996 \end{bmatrix},$$

$$K_{0,fin} = \begin{bmatrix} 0.2919 \end{bmatrix}, \quad K_{1,fin} = \begin{bmatrix} -1.8203 \end{bmatrix}, \quad K_{2,fin} = \begin{bmatrix} 0.1258 \end{bmatrix}.$$

Table 1: Performance of the HCG1 method on Example 5.3

k	$J(K_0,K_1,K_2)$	$\Vert \nabla J(K_0,K_1,K_2) \Vert$	$\rho(A(K_0,K_1,K_2))$
0	3.1581e+002	3.7124e+002	9.07e−001
1	1.6734e+002	7.6055e+001	5.79e−001
2	1.6543e+002	1.3828e+002	8.23e−001

3	1.6285e+002	8.6228e+001	9.08e−001
⋮	⋮	⋮	⋮
23	1.4220e+002	1.3181e−004	7.38e−001
24	1.4220e+002	8.2648e−005	7.38e−001

Table 1: shows the convergence behavior of the method HCG1 to reach the stationary point of the optimization problem (8) when $d = 3$.

Example 5.4 This test problem is borrowed from [8] having the following data matrices **with period $d = 3$**:

$$A_0 = \begin{bmatrix} 1.1018 & 0.0557 & 0.0503 \\ 0.0253 & 1.0938 & 0.0028 \\ 0.0676 & 0.0845 & 1.0871 \end{bmatrix}, A_1 = \begin{bmatrix} 1.0069 & 0.0012 & 0.0204 \\ 0.0378 & 1.0147 & 0.0616 \\ 0.0827 & 0.0205 & 1.0288 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1.0903 & 0.0433 & 0.0761 \\ 0.0037 & 1.0878 & 0.0456 \\ 0.0721 & 0.0544 & 1.0341 \end{bmatrix}, B_0 = \begin{bmatrix} 0.0498 & 0.0990 \\ 0.0655 & 0.0786 \\ 0.0863 & 0.0248 \end{bmatrix}, B_1 = \begin{bmatrix} 0.0208 & 0.0457 \\ 0.0047 & 0.0959 \\ 0.0769 & 0.0505 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 0.0228 & 0.0330 \\ 0.0214 & 0.0567 \\ 0.0703 & 0.0177 \end{bmatrix}, C_0 = \begin{bmatrix} 0.4100 \\ 0.9400 \\ 0.9200 \end{bmatrix}, C_1 = \begin{bmatrix} 0.4200 \\ 0.8500 \\ 0.5300 \end{bmatrix}, C_2 = \begin{bmatrix} 0.7000 \\ 0.3800 \\ 0.8600 \end{bmatrix},$$

$$Q_0 = Q_1 = Q_2 = I_3, \quad R_0 = R_1 = R_2 = I_2, \quad V = I_3.$$

The uncontrolled system is not discrete-time Schur stable, where $\rho(A_2A_1A_0) = 1.5167$. Starting with the following initial $K_{0,0}, K_{1,0}, K_{2,0} \in \mathcal{D}_d$ the methods HCG1, PRP require 52 and 77 iterations with CPU times 0.68 and 0.87, respectively, while the method HCG2 failed to converge the stationary point $K_{0,fin}, K_{1,fin}, K_{2,fin}$. The starting and final feedback gain matrices are:

$$K_{0,0} = \begin{bmatrix} 0.9021 \\ 2.7905 \end{bmatrix}, \quad K_{1,0} = \begin{bmatrix} -8.8787 \\ -5.9218 \end{bmatrix}, \quad K_{2,0} = \begin{bmatrix} -21.2334 \\ 0.0641 \end{bmatrix},$$

$$K_{0,fin} = \begin{bmatrix} -1.4173 \\ 3.6178 \end{bmatrix}, \quad K_{1,fin} = \begin{bmatrix} 2.1674 \\ -4.2130 \end{bmatrix}, \quad K_{2,fin} = \begin{bmatrix} -7.3428 \\ -5.1525 \end{bmatrix}.$$

V. Conclusion

The main goal of this research work is to study the performance of some efficient numerical optimization methods for tackling optimal control problem, namely the static output feedback design problem. The related problem of the static output feedback design for periodic systems is one of the most important problems in modern control. For this problem two hybrid conjugate gradient methods are proposed to find its local solution of the corresponding optimization problem. Global convergence is established for the hybrid conjugate gradient method. It is important to point out that the numerical method for solving the SOF problem requires a starting feasible point with respect to an eigenvalue constraint. Such a feasible point can be easily obtained by any of the considered solvers of the eigenvalue assignment problem. All methods considered in the paper are tested on wide range of test problems from the literature.

References

[1]. F.A. Aliev and V.B. Larin, Optimization problems for periodic systems, International Applied Mechanics, **45** (2009), 1162–4511

[2]. N. Andrei, Hybrid conjugate gradient algorithm for unconstrained optimization. J. Optim. Theory Appl., **141(2)** (2009), 249–264.

[3]. S. Bittanti and P. Colaneri.(2009) Periodic Systems Filtering and Control, Springer, Verlag London Limited.

[4]. P. Colaneri. Periodic control systems: Theoretical aspects, Applied and computational mathematics, **3**:2 (2004), 84–-94.

[5]. Y. Chen, Global convergence of a new conjugate gradient method with wolfe type line search, Journal of Information and Computing Science, **7** (2012), 067–071.

[6]. Y.H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optimiz., **10** (1999), 177–182.

[7]. Y.H. Dai and Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res., **103**, (2001), 33–47

[8]. C. Farges, D. Peaucelle and D. Arzelier, Resilient static output feedback stabilization of linear periodic systems. In: 5th IFAC Symposium on Robust Control Design. Toulouse, (2006), 5–7.

[9]. R. Fletcher and C. Reeves, Function minimization by conjugate gradients, Computer Journal, **7** (1964), 149–154.

[10]. M.R. Hestenes and E.L Stiefel, Methods of conjugate gradients for solving linear systems, Journal Res. Natl. Bur. Stand., **49** (1952), 409–436.

[11]. J. Jian, L. Han, X. Jiang, A hybrid conjugate gradient method with descent property for unconstrained optimization, Applied Mathematical Modelling, **39** (2015), 1281–1290

[12]. S.B. Kafaki and R. Ghanbari, A hybridization of the Polak-Ribière-Polyak and Fletcher–Reeves conjugate gradient methods, Numer Algor, **68** (2015), 481–495.

[13]. M. Kočvara, F. Leibfritz, M. Stingl and D. Henrion, A nonlinear SDP algorithm for static output feedback problems in COMPlib, in Proceedings of the 16th IFAC World Congress on Automatic Control, Prague, 2005, Elsevier, ISBN: 0-08-045108-X.

[14]. V.B. Larin, On static output–feedback stabilization of a periodic system, International Applied Mechanics, **42** (2006), 3.

[15]. J. Lavaei and Amir G. Aghdam, Simultaneous LQ Control of a Set of LTI Systems using Constrained Generalized Sampled-Data Hold Functions, This work has been supported by the Natural Sciences and Engineering Research Council of Canada under grant RGPIN-262127-03.

[16]. I.E. Livieris, V. Tampakas and P. Pintelas, A descent hybrid conjugate gradient method based on the memoryless BFGS update, Numer Algor, https://doi.org/10.1007/s11075-018-0479-1.

[17]. 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.

[18]. 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.

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

[20]. E.M.E. Mostafa. A conjugate gradient method for discrete–time output feedback control design, Journal of computational mathematics, **30** (2012), 279–297.

[21]. E.M.E. Mostafa. Computational design of optimal discrete–time output feedback controller, Journal of the Operations Research Society of Japan, **51** (2008), 15–28.

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

[23]. 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.

[24]. E.M.E. Mostafa and F.F.S. Omar. On the solution of the eigenvalue assignment Problem by derivative–free optimization methods, AMO–Advanced Modeling and Optimization, **30** (2018), 371–394

[25]. J. Nocedal and S.J. Wright.(1999), Numerical Optimization, Springer Series in Operations Research, Springer, New York, Berlin.

[26]. T. Rautert and E.W. Sachs, Computational design of optimal output feedback controllers, SIAM J. Optim., **7** (1997), 837–852.

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

[28]. A. Varga, Computation of Kalman decompositions of periodic systems. European Journal of Control, **10** (2004a), 1– –8.

[29]. Y. Yang. An efficient LQR design for discrete–time linear periodic system based on a novel lifting method, Automatica, **87** (2018), 383–388.

[30]. G. Zoutendijk, Nonlinear programming, computational methods, in Integer and Nonlinear Programming, J. Abadie, ed., North–Holland, Amsterdam, (1970), 37–86.

Appendix

Lemma 8.1 *Let K_0 and $K_1 \in \mathcal{D}_d$. It holds that*

$$Tr(U(K_0,K_1)V) = Tr(P(K_0,K_1)L(K_0,K_1)),$$

where the matrix variable $P(K_0,K_1) = P(\cdot)$ solves the following discrete Lyapunov equation:

$$P(\cdot) = A(K_0,K_1)P(\cdot)A(K_0,K_1)^T + V,$$

Proof: From the trace properties we can show that

$$\begin{aligned} Tr(U(K_0,K_1)V) &= Tr(U(K_0,K_1)(P(\cdot) - A(K_0,K_1)P(\cdot)A(K_0,K_1)^T)) \\ &= Tr(U(K_0,K_1)P(\cdot)) - Tr(U(K_0,K_1)A(K_0,K_1)P(\cdot)A(K_0,K_1)^T) \\ &= Tr(P(\cdot)U(K_0,K_1)) - Tr(P(\cdot)A(K_0,K_1)^T U(K_0,K_1)A(K_0,K_1)) \\ &= Tr(P(\cdot)(U(K_0,K_1) - A(K_0,K_1)^T U(K_0,K_1)A(K_0,K_1))) \\ &= Tr(P(\cdot)L(K_0,K_1)). \end{aligned}$$

For the Example 5.3 and Example 5.4 we presented the next lemma follows, let $d = 3$ we obtained the objective function:

$$\min_{K_0,K_1,K_2 \in \mathcal{D}_d} J(K_0,K_1,K_2) = Tr(L(K_0,K_1,K_2)P(K_0,K_1,K_2)), \tag{53}$$

where the matrix

$$L(K_0,K_1,K_2) = \overline{Q}_0 + \overline{A}_0^T \overline{Q}_1 \overline{A}_0 + \overline{A}_1^T \overline{A}_0^T \overline{Q}_2 \overline{A}_0 \overline{A}_1 \tag{54}$$

and the matrix $P(\cdot) = P(K_0,K_1,K_2)$ is a solution of the discrete Lyapunov equation:

$$P(\cdot) = \psi(\cdot)P(\cdot)\psi(\cdot)^T + V, \tag{55}$$

where

$$\psi(\cdot) = \overline{A}_2 \overline{A}_1 \overline{A}_0$$

gives the first–order directional derivative of the objective function $J(K_0,K_1,K_2)$.
The next Lemma provides a discrete Lyapunov equation required for obtaining the gradient of objective function (53) .

Lemma 8.2 *Consider the optimization problem* (53) - (55) *, where K_0, K_1 and $K_2 \in \mathcal{D}_d$. Then $P(K_0,K_1,K_2)$ defined by* (55) *is differentiable and directional derivatives $\Delta P(\cdot)\Delta K_0, \Delta P(\cdot)\Delta K_1$* and $\Delta P(\cdot)\Delta K_2$ of $P(K_0,K_1,K_2)$ are given by the discrete Lyapunov equations:

$$\Delta P(\cdot)\Delta K_0 = \psi(\cdot)\Delta P(\cdot)\Delta K_0 \psi(\cdot)^T + \psi(\cdot)P(\cdot)C_0^T \Delta K_0^T B_0^T \overline{A}_1^T \overline{A}_2^T + \overline{A}_2 \overline{A}_1 B_0 \Delta K_0 C_0 P(\cdot)\psi(\cdot)^T, \tag{56}$$

$$\Delta P(\cdot)\Delta K_1 = \psi(\cdot)\Delta P(\cdot)\Delta K_1\psi(\cdot)^T + \psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T B_1^T\overline{A}_2^T + \overline{A}_2 B_1\Delta K_1 C_1\overline{A}_0 P(\cdot)\psi(\cdot)^T, \quad (57)$$
$$\Delta P(\cdot)\Delta K_2 = \psi(\cdot)\Delta P(\cdot)\Delta K_2\psi(\cdot)^T + \psi(\cdot)P(\cdot)\overline{A}_0^T\overline{A}_1^T C_2^T\Delta K_2^T B_2^T + B_2\Delta K_2 C_2\overline{A}_1\overline{A}_0 P(\cdot)\psi(\cdot)^T, \quad (58)$$
where $\psi(\cdot) = \psi(K_0, K_1, K_2)$.

Proof: The directional derivatives of (55) with respect to K_0, K_1 and K_2 respectively are given

$$\begin{aligned}\Delta P(\cdot)\Delta K_0 &= \psi(\cdot)P(\cdot)C_0^T\Delta K_0^T B_0^T\overline{A}_1^T\overline{A}_2^T + \left[\psi(\cdot)\Delta P(\cdot)\Delta K_0 + \overline{A}_2\overline{A}_1 B_0\Delta K_0 C_0 P(\cdot)\right]\psi(\cdot)^T\\ &= \psi(\cdot)\Delta P(\cdot)\Delta K_0\psi(\cdot)^T + \psi(\cdot)P(\cdot)C_0^T\Delta K_0^T B_0^T\overline{A}_1^T\overline{A}_2^T + \overline{A}_2\overline{A}_1 B_0\Delta K_0 C_0 P(\cdot)\psi(\cdot)^T,\end{aligned}$$

$$\begin{aligned}\Delta P(\cdot)\Delta K_1 &= \psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T B_1^T\overline{A}_2 + \left[\psi(\cdot)\Delta P(\cdot)\Delta K_1 + \overline{A}_2 B_1\Delta K_1 C_1\overline{A}_0 P(\cdot)\right]\psi(\cdot)^T\\ &= \psi(\cdot)\Delta P(\cdot)\Delta K_1\psi(\cdot)^T + \psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T B_1^T\overline{A}_2^T + \overline{A}_2 B_1\Delta K_1 C_1\overline{A}_0 P(\cdot)\psi(\cdot)^T,\end{aligned}$$

$$\begin{aligned}\Delta P(\cdot)\Delta K_2 &= \psi(\cdot)P(\cdot)\overline{A}_0^T\overline{A}_1^T C_2^T\Delta K_2^T B_2^T + \left[\psi(\cdot)\Delta P(\cdot)\Delta K_2 + B_2\Delta K_2 C_2\overline{A}_1\overline{A}_0 P(\cdot)\right]\psi(\cdot)^T\\ &= \psi(\cdot)\Delta P(\cdot)\Delta K_2\psi(\cdot)^T + \psi(\cdot)P(\cdot)\overline{A}_0^T\overline{A}_1^T C_2^T\Delta K_2^T B_2^T + B_2\Delta K_2 C_2\overline{A}_1\overline{A}_0 P(\cdot)\psi(\cdot)^T\end{aligned}$$

□

The next lemma gives the first–order directional derivative of the objective function $J(K_0, K_1, K_2)$.

Lemma 8.3 *Let $d = 3$ and $K_0, K_1, K_2 \in \mathcal{D}_d$. The gradient of the objective function* (53) $J(\cdot)$ *is given by*

$$\begin{aligned}J_{K_0}(\cdot)\Delta K_0 &= 2Tr\left(\left[R_0K_0C_0 + B_0^T\left(\overline{Q}_1\overline{A}_0 + \overline{Q}_2\overline{A}_0\overline{A}_1\overline{A}_1^T + \overline{A}_1^T\overline{A}_2^T U\psi\right)\right]PC_0^T\Delta K_0^T\right),\\ J_{K_1}(\cdot)\Delta K_1 &= 2Tr\left(\left[R_1K_1C_1\overline{A}_0\overline{A}_0^T P + B_1^T\left(\overline{A}_0^T\overline{Q}_2\overline{A}_0\overline{A}_1 P + \overline{A}_2^T U\psi P\overline{A}_0^T\right)\right]C_1^T\Delta K_1^T\right),\\ J_{K_2}(\cdot)\Delta K_2 &= 2Tr\left(\left[R_2K_2C_2\overline{A}_0\overline{A}_1 P\overline{A}_1^T\overline{A}_0^T + B_2^T U\psi P\overline{A}_0^T\overline{A}_1^T\right]C_2^T\Delta K_2^T\right),\\ U &= \psi^T U\psi + L(\cdot),\end{aligned} \quad (59)$$

where P and U solve the discrete Lyapunov equations (55) and (59) , respectively.

Proof: By differentiating the objective function with respect to K_0 in the direction of ΔK_0,

$$\begin{aligned}J_{K_0}(\cdot)\Delta K_0 &= Tr(L(\cdot)\Delta P(\cdot)\Delta K_0) + Tr(\Delta L(\cdot)\Delta K_0 P(\cdot))\\ &= Tr(L(\cdot)\Delta P(\cdot)\Delta K_0) + 2Tr\left((R_0K_0C_0 + B_0^T\overline{Q}_1\overline{A}_0 + B_0^T\overline{Q}_2\overline{A}_0\overline{A}_1\overline{A}_1^T\right)PC_0^T\Delta K_0^T.\end{aligned}$$

From the Lyapunov equation (55) and (56) we have

$$\begin{aligned}Tr(L(\cdot)\Delta P(\cdot)\Delta K_0) &= Tr\left(U(\psi PC_0^T\Delta K_0^T B_0^T\overline{A}_1^T\overline{A}_2^T)\right) + Tr\left(U(\cdot)(\overline{A}_2\overline{A}_1 B_0\Delta K_0 C_0 P(\cdot)\psi(\cdot)^T)\right)\\ &= 2Tr\left(B_0^T\overline{A}_1^T\overline{A}_2^T U\psi PC_0^T\Delta K_0^T\right).\end{aligned}$$

Hence, the directional derivative of the objective function in the direction of ΔK_0 is

$$J_{K_0}(\cdot)\Delta K_0 = 2Tr\left(\left(R_0K_0C_0 + B_0^T\left(\overline{Q}_1\overline{A}_0 + \overline{Q}_2\overline{A}_0\overline{A}_1\overline{A}_1^T + \overline{A}_1^T\overline{A}_2^T U\psi\right)\right)PC_0^T\Delta K_0^T\right).$$

By differentiating the objective function with respect to K_1 in the direction of ΔK_1,

$$\begin{aligned}J_{K_1}(\cdot)\Delta K_1 &= Tr(L(\cdot)\Delta P(\cdot)\Delta K_1) + Tr(\Delta L(\cdot)\Delta K_1 P(\cdot))\\ &= Tr(L(\cdot)\Delta P(\cdot)\Delta K_1) + 2Tr\left(\left(R_1K_1C_1\overline{A}_0\overline{A}_0^T + B_1^T\overline{A}_0^T\overline{Q}_2\overline{A}_0\overline{A}_1\right)PC_1^T\Delta K_1^T\right).\end{aligned}$$

From the Lyapunov equation (55) and (56) we have

$$\begin{aligned}Tr(L(\cdot)\Delta P(\cdot)\Delta K_1) &= Tr\left(U\psi(\cdot)P(\cdot)\overline{A}_0^T C_1^T\Delta K_1^T B_1^T\overline{A}_2^T\right) + Tr\left(U\overline{A}_2 B_1\Delta K_1 C_1\overline{A}_0 P(\cdot)\psi(\cdot)^T\right)\\ &= 2Tr\left(B_1^T\overline{A}_2^T U\psi P\overline{A}_0 C_1^T\Delta K_1^T\right).\end{aligned}$$

Hence, the directional derivative of the objective function in the direction of ΔK_1 is

$$J_{K_1}(\cdot)\Delta K_1 = 2Tr\left(\left(R_1K_1C_1\overline{A}_0\overline{A}_0^T P + B_1^T\left(\overline{A}_0\overline{Q}_2\overline{A}_0\overline{A}_1 P + \overline{A}_2^T U\psi P\overline{A}_0^T\right)\right)C_1^T\Delta K_1^T\right).$$

By differentiating the objective function with respect to K_2 in the direction of ΔK_2,

$$\begin{aligned}J_{K_2}(\cdot)\Delta K_2 &= Tr(L(\cdot)\Delta P(\cdot)\Delta K_2) + Tr(\Delta L(\cdot)\Delta K_2 P(\cdot))\\ &= Tr(L(\cdot)\Delta P(\cdot)\Delta K_2) + 2Tr\left(R_2K_2C_2\overline{A}_0\overline{A}_1 P\overline{A}_1^T\overline{A}_0^T C_2^T\Delta K_2^T\right).\end{aligned}$$

From the Lyapunov equation (55) and (56) we have

$$\begin{aligned}Tr(L(\cdot)\Delta P(\cdot)\Delta K_2) &= Tr\left(U\psi(\cdot)P\overline{A}_0^T\overline{A}_1^T C_2^T\Delta K_2^T B_2^T\right) + Tr\left(UB_2\Delta K_2 C_2\overline{A}_1\overline{A}_0 P\psi(\cdot)^T\right)\\ &= 2Tr\left(B_2 U\psi P\overline{A}_0^T\overline{A}_1^T C_2^T\Delta K_2\right).\end{aligned}$$

Hence, the directional derivative of the objective function in the direction of ΔK_2 is

$$J_{K_2}(\cdot)\Delta K_2 = 2Tr\left(\left[R_2 K_2 C_2 \overline{A}_0 \overline{A}_1 P \overline{A}_1^T \overline{A}_0^T + B_2^T U \psi P \overline{A}_0^T \overline{A}_1^T\right] C_2^T \Delta K_2^T\right).$$

Fatma F.S. Omar. "A Hybrid Conjugate Gradient Method for Discrete–Time Periodic Static Output Feedback Control Design." *IOSR Journal of Mathematics (IOSR-JM)*, 17(3), (2021): pp. 12-26.