Proc. 41th IEEE Conference on Decision and Control,
pp. 3156-3161, Las Vegas, NV, USA, 2002

A Bundle Method for Solving the Fixed Order Control Problem

Max Rotunno[1] Raymond A. de Callafon[2]

Abstract

Recently bundle methods have been proposed for solving Semi-Definite Programming (SDP) problems which consist of minimizing a linear objective subject to Linear Matrix Inequality (LMI) constraints. These methods are appealing especially for large scale problems. Many control problems can be cast as SDPs. In contrast fixed order control problems in the LMI framework require an additional rank constraint on the LMIs which makes the problem insolvable with traditional SDP methods. One way of imposing the rank constraint is to formulate the fixed order control problem in terms of the eigenvalues of the LMIs. This makes the problem nondifferentiable, but amenable to bundle methods.

1 Introduction

Many control problems can be formulated as minimizing a linear objective subject to Linear Matrix Inequalities (LMI) [15]. This is known as Semidefinite Programming (SDP) and can be written as,

$$\begin{aligned} &\min_y -b^T y \\ &\text{s.t.} \ \ -C + \sum_{j=1}^{n} y^j A_j = Z(y) \\ &\qquad Z(y) \preceq 0, \end{aligned} \tag{1}$$

where $y \in \mathbb{R}^n$, $Z(y)$ is an affine mapping from $\mathbb{R}^n$ to the space $y \in \mathbb{S}^m$ of $m \times m$ symmetric matrices and $\preceq$ means that the left hand side is negative semidefinite. In general for control applications $Z(y)$ has block diagonal structure which can be easily exploited.

SDPs are considered tractable because they are convex programs and there are polynomial-time algorithms to solve them. However, these algorithms are generally expensive, since the number of decision variables is not confined to only y, and become prohibitive for large scale problems. This has motivated new research efforts such as [7, 12, 13] that try to solve (1) directly by replacing the negative semidefinite constraint with the equivalent constraint that the maximum eigenvalue of $Z(y)$ be less than or equal to zero. While these methods have the advantage that the number of decision variables during the optimization is greatly reduced, their main drawback is that the maximum eigenvalue is not differentiable whenever its multiplicity is not unity. For this reason the above methods are based on nonsmooth optimization methods.

The fixed order control problem is one in which the number of states in the controller is fixed. This problem can be formulated from (1) by adding a rank constraint to certain submatrices present in $Z(y)$ and can be expressed as:

$$\begin{aligned} &\min_y -b^T y \\ &\text{s.t.} \qquad Z(y) \preceq 0 \\ &\qquad \text{Rank}\, P(y) \leq r \end{aligned} \tag{2}$$

where $P(y)$ is $q \times q$ submatrix of $Z(y)$. The addition of the rank constraint makes the fixed order control problem nonlinear and non convex. Methods of solution of the rank constraint problem have been proposed in [14, 3, 1, 6].

Since $RP(y)$ is symmetric, the rank constraint is equivalent to requiring that the smallest $k = q - r$ eigenvalues of $P(y)$ be zero. From a numerical point of view the rank of a matrix is not completely well defined. In many cases, such as in model reduction, one decides the rank of a matrix when the k^{th} smallest eigenvalue (singular value) is sufficiently smaller than the $k^{th} + 1$ eigenvalue (singular value). So in many cases enforcing the k smallest eigenvalues to be exactly zero is not necessary and could actually lead to conservative results. In [14] these facts are used to come up with a method for synthesizing fixed order controllers by constraining the k smallest eigenvalues to be sufficiently small. The main drawback of [14] is the use of a general purpose constrained smooth optimization routine to solve a nonsmooth problem.

This paper builds on the method presented in [14] in that the solution is found using a constrained nonsmooth bundle type optimization routine which is based on [10]. The outline of the paper is as follows. First, an example of a fixed order controller problem in terms of

[1]Dept. of Mechanical and Aerospace Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla CA 92093-0411, U.S.A. e-mail: mrotunno@ucsd.edu
[2]e-mail: callafon@ucsd.edu

LMIs is given for the case of $\mathcal{H}_\infty$ optimal control case. The fixed order controller problem is then cast as a nonlinear programming problem in which eigenvalues of certain matrices appear.

Section 3 gives the necessary theory on the nonsmooth optimization is given. Subsequently, the algorithm is given. Finally, conclusions are presented in Section 5.

The following notation will be used throughout the paper. $\mathbb{S}^m$ denotes the Hilbert space of real $m \times m$ symmetric matrices equipped with the inner product $\langle X, Y \rangle = \mathrm{tr}(XY)$ where $\mathrm{tr}(A)$ is the trace of A. The eigenvalues of a matrix $A \in \mathcal{S}^m$ are ordered $\lambda_1(A) \geq \cdots \geq \lambda_m(A)$. The LMI function $Z : \mathbb{R}^n \to \mathbb{S}^m$ is affine with its components given by $Z(x) = -C + \sum_{k=1}^n x_k A_k$. The function $\bar{\lambda}(x) : \mathbb{R}^n \to \mathbb{R}$ defined by $\bar{\lambda} := \lambda_1(Z(x))$ is particularly important since $Z(x) \preceq 0$ if and only if (iff) $\bar{\lambda}(x) \leq 0$.

2 Fixed Order $\mathcal{H}_\infty$ Controllers

Let a minimal realization of a linear time invariant (LTI) discrete time plant P be given by the following state space realization:

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 w(k) + B_2 u(k) \\ z(k) &= C_1 x(k) + D_{11} w(k) + D_{12} u(k) \\ y(k) &= C_2 x(k) + D_{21} w(k) \end{aligned} \quad (3)$$

Where $A \in \mathbb{R}^{n \times n}$, (A, B_2, C_2) is stabilizable and detectable and k is used to denote the sample instant. Let the servo controller C to be designed be given by the following minimal (LTI) state space realization

$$\begin{aligned} x_c(k+1) &= A_c x_c(k) + B_c y(k) \\ u(k) &= C_c x_c(k) + D_c y(k) \end{aligned} \quad (4)$$

where $A_c \in \mathbb{R}^{n_c \times n_c}$ and n_c is the order of the controller to be designed.

The closed-loop transfer function T_{wz} from w to z is obtained as

$$T_{wz} = D_{cl} + C_{cl}(zI - A_{cl})^{-1} B_{cl} \quad (5)$$

where

$$A_{cl} = \begin{pmatrix} A + B_2 D_c C_2 & B_2 C_c \\ B_c & A_c \end{pmatrix} \quad B_{cl} = \begin{pmatrix} B_1 + B_2 D_c D_{21} \\ B_c D_{21} \end{pmatrix}$$
$$C_{cl} = \begin{pmatrix} C_1 + D_{12} D_c C_2 & D_{12} C_c \end{pmatrix} D_{cl} = D_{11} + D_{12} D_c D_{21}$$

By means of the Bounded Real Lemma [2], the closed-loop transfer function T_{wz} satisfies an $\mathcal{H}_\infty$-norm performance bound $\|T_{wz}\| < \gamma$ if and only if A_{cl} is stable in the discrete time sense ($|\lambda_i(A_{cl})| < 1$) and the matrix inequality

$$\begin{pmatrix} -X_{cl}^{-1} & A_{cl} & B_{cl} & 0 \\ A_{cl}^T & -X_{cl} & 0 & C_{cl}^T \\ B_{cl}^T & 0 & -\gamma I & D_{cl}^T \\ 0 & C_{cl} & D_{cl} & -\gamma I \end{pmatrix} < 0 \quad (6)$$

holds for some $X_{cl} > 0$.

For a given value of $\gamma > 0$ finding a controller C for which the expression (6) holds leads to the so called suboptimal $\mathcal{H}_\infty$ control design. The optimal $\mathcal{H}_\infty$ control design problem consists instead in finding the minimum value of $\gamma > 0$ such that (6) holds. The existence of $\mathcal{H}_\infty$ optimal or suboptimal controllers of order n_c is fully characterized by the following result [2].

Theorem 1 *Let $\mathcal{N}_x$ and $\mathcal{N}_y$ denote orthonormal bases of the null spaces of (B_2^T, D_{12}^T) and (C_2, D_{21}), respectively. There exists a controller of order n_c which stabilizes the system and yields $\|T_{wz}\|_\infty < \gamma$ if and only if*

$$M := \tilde{\mathcal{N}}_x^T \begin{pmatrix} AXA^T - X & AXC_1^T & B_1 \\ C_1 X A^T & C_1 X C_1^T - \gamma I & D_{11} \\ B_1^T & D_{11}^T & -\gamma I \end{pmatrix} \tilde{\mathcal{N}}_x < 0, \quad (7)$$

$$N := \tilde{\mathcal{N}}_y^T \begin{pmatrix} A^T Y A - Y & A^T Y B_1 & C_1^T \\ B_1^T Y A & B_1^T Y B_1 - \gamma I & D_{11}^T \\ C_1 & D_{11} & -\gamma I \end{pmatrix} \tilde{\mathcal{N}}_y < 0, \quad (8)$$

$$P := \begin{pmatrix} X & I \\ I & Y \end{pmatrix} \geq 0, \quad (9)$$

where

$$\tilde{\mathcal{N}}_x^T = \begin{pmatrix} \mathcal{N}_x & 0 \\ 0 & I \end{pmatrix} \text{ and } \tilde{\mathcal{N}}_y^T = \begin{pmatrix} \mathcal{N}_y & 0 \\ 0 & I \end{pmatrix}.$$

Furthermore the following rank condition has to be satisfied,

$$\mathit{Rank} \begin{pmatrix} X & I \\ I & Y \end{pmatrix} \leq n + n_c. \quad (10)$$

Proof: See [2].

For a fixed value of γ the constraints (7)–(9) are Linear Matrix Inequalities (LMI) and define a convex set in the variables (X, Y). The suboptimal $\mathcal{H}_\infty$ problem with performance γ is solvable if and only if this set is non empty. It should be noted that the rank constraint is satisfied trivially when $n_c = n$, which corresponds to the design of a controller that has the same order as the plant P. This case will be denoted in this paper by the full order case or full order controller. In the design of a reduced order controller or reduced order case, the rank constraint makes the problem non convex, so that efficient semi–definite programming (SDP) techniques are not applicable and other methods are needed.

Given any solution (γ, X, Y) of (7)–(10), it is possible to construct a matrix X_{cl} [2]. Then a $\mathcal{H}_\infty$ controller of order n_c can be computed by solving the Bounded Real Lemma inequality (6) for the controller data (A_c, B_c, C_c, D_c). This problem can be solved using SDP.

2.1 Solution via Numerical Optimization

From Theorem 1 above, it is evident that $\mathcal{H}_\infty$ controllers of order $n_c < n$ can be obtained if one finds a trio (γ, X, Y) so that the LMI's (7)–(9) and the rank constraint (10) are satisfied. Since the matrix P is symmetric its rank can be determined directly from its eigenvalues: $\lambda_1(P) > \lambda_2(P) > \ldots > \lambda_k(P) > \lambda_{k+1}(P) > \ldots > \lambda_{2n}(P)$.

Since P is of order $2n$, the rank constraint (10) is equivalent to requiring that the $k = n - n_c$ smallest eigenvalues of P be zero. From a numerical point of view rounding errors and fuzzy data make rank determination a nontrivial exercise. Following [5] we will assume P as having rank $n + nc$ if $\lambda_{k+1} > \delta \geq \lambda_k$ where the value of δ is chosen based upon the precision with which data is known.

The optimal fixed order $\mathcal{H}_\infty$ controller can be found by solving the following optimization problem:

$$\begin{aligned} \min_{\gamma, X, Y} \; & \gamma \\ \text{s.t.} \quad & \lambda_1(M(\gamma, X)) < 0 \\ & \lambda_1(N(\gamma, Y)) < 0 \\ & \lambda_1(-P(X, Y)) < 0 \\ & \lambda_k(P(X, Y)) \leq \delta \end{aligned} \tag{11}$$

The following facts should be noted:

- The last two constraints in (11) effectively impose the rank constraint: requiring P to be positive definite and requiring that the eigenvalue $\lambda_k(P)$ be sufficiently small, forces the k smallest eigenvalues of P towards zero.
- By forcing P to be rank deficient at the optimal point makes the problem nonsmooth since P will have k eigenvalues equal to zero and thus have a multiplicity of k.
- (11) can be cast as (2) by defining the block diagonal matrix,

  $$Z = \text{blkdiag}\left(M(\gamma, X), N(\gamma, Y), -P(X, Y)\right).$$

- The numerical value of δ can be chosen as $\delta = 10^p \|P(X_0, Y_0\|$ where p is the precision with which the data is known and (X_0, Y_0) are the initial values with which the optimization routine is started.

A initial point can be obtained by first solving the following semi-definite problem for a fixed value of γ:

$$\begin{aligned} \min_{X,Y} \quad & \text{tr}(X + Y) \\ \text{s.t.} \quad & (7) - (9) \end{aligned}$$

This minimization is guaranteed [4] to give a controller of order at least $n - 1$, and generally does better.

In [14] (11) was solved using a standard smooth constrained optimization code, even though the minimization problem (11) is non-differentiable. Results reported in [14] were encouraging even though the number of iterations and function evaluations tended to be high, thus making the method inefficient.

3 Bundle Methods for Constrained Nonsmooth Optimization

Bundle methods were originally developed in the context of unconstrained nonsmooth convex optimization. They were then further extended to constrained and non-convex problems [9]. The method proposed in this section is based on [9] and [11].

In this section the optimization problem (11) is rewritten as:

$$\begin{aligned} \min_{x} \quad & b^T x \\ \text{s.t.} \quad & c(x) \leq 0 \end{aligned} \tag{12}$$

where $x = \text{vec}(\gamma, X, Y) \in \mathbb{R}^n$ i.e. all the unknowns are stored in a vector and

$$\begin{aligned} c(x) = \max(&\lambda_1(M(\gamma, X)), \lambda_1(N(\gamma, Y) \\ &\lambda_1(-P(X, Y)), \lambda_k(P(X, Y)) - \delta) \end{aligned}$$

We will assume that for each $x \in \mathbb{R}^n$ we can compute an arbitrary subgradient $g_c(x) \in \partial c(x)$ and that the generalized Slater constraint qualification holds: $c(\tilde{x}) \leq 0$ for some $\tilde{x} \in \mathbb{R}^n$. In these terms the optimization problem is nonsmooth due to the fact that the constraint $c(x)$ involves the max function and that at any given x the eigenvalues of the matrices involved may have a multiplicity greater than one. In this case one has to consider the subdifferential $\partial c(x)$ of $c(x)$ at x.

The algorithm to be described will generate a sequence of points $x_1, x_2, \cdots$, search directions $d_1, d_2, \cdots$ and stepsizes $t_L^1, t_L^2, \cdots$ in $(0, 1]$, related by $x_{k+1} = x_k + t_L^k d_k$ for $k = 1, 2, \cdots$, where x_1 is the given staring point. The method will also calculate a sequence of trial points $y_{k+1} = x_k + t_R^k d^k$ for $k = 1, 2, \cdots$ and subgradients $g_c^k = g_c(y^k)$ of the constraint $c(x)$ for $k \geq 1$, where $y_1 = x_1$.

Each trial point y_j will define a linearization of c

$$\ell_c^j(x) = c(y_j) + \langle g_c^j, x - y_j \rangle \qquad \forall x \tag{13}$$

At the k^{th} iteration, we shall have a subset J_c^k of $\{1, \cdots, k\}$ and the corresponding linearizations ℓ_c^j, $j \in J_c^k$, given by the $N+1$ vectors (g_c^j, c_j^k) in the form

$$\ell_c^j(x) = c_k^j + \langle g_c^j, x - x_k \rangle \qquad \forall x \tag{14}$$

where $c_j^k = \ell_c^j(x_k)$. The above form will enable us not to store the points y_j. The available linearizations will define a polyhedral approximation of $c(x)$

$$\ell_c^k(x) = \max(c_j^k + \langle g_c^j, x - x_k \rangle : j \in J_c^k) \qquad \forall x \tag{15}$$

If the constraint $c(x)$ in (12) is replaced by its polyhedral approximation ℓ_c^k in (15) we obtain

$$\begin{aligned} \min_x \quad & b^T x \\ \text{s.t.} \quad & \ell_c^k(x) \leq 0 \end{aligned} \tag{16}$$

which may be regarded as a local approximation to the original problem (12). However the above problem may have no finite solutions. Therefore we shall choose the next trial point y_{k+1} to satisfy

$$\begin{aligned} \min_y \quad & b^T y + \frac{1}{2}\|y - x_k\|_{B_k}^2 \\ \text{s.t.} \quad & \ell_c^k(y) \leq 0 \end{aligned} \tag{17}$$

where the regularizing $\frac{1}{2}\|y - x_k\|_{B_k}^2$ term is introduced so as to keep y_{k+1} in the region where $\ell_c^k(x)$ is a close approximation to $c(x)$ and $\|x\|_{B_k} = (x^T B_k x)^{1/2}$ is an ellipsoidal norm, where B_k is a positive-definite symmetric matrix. The direction $d_k = y_{k+1} - x_k$ can be found by solving the following quadratic programming problem

$$\begin{aligned} \min_y \quad & b^T d + \frac{1}{2} d^T B_k d \\ \text{s.t.} \quad & c_j^k + \langle g_c^j, d \rangle \leq 0 \quad j \in J_c^k \end{aligned} \tag{18}$$

In order to guarantee global convergence of the method it is necessary to define a merit function so as to evaluate if a current iterate is better than the previous. In the case of problem (18) a natural merit function is given by $P_\infty(x, \rho) = b^T x + \rho \max(c(x), 0)$ the exact L_∞ penalty function associated with problem (12). This can be seen by Lemma 2.1 in [10] where it is shown that if d solves (18) then it also solves

$$\min_d \hat{P}_\infty(d, \rho_k) + \frac{1}{2} d^T B_k d \tag{19}$$

where

$$\rho_k \geq \sum_{j \in J_c^k} \lambda_k^j$$

$$\hat{P}_\infty(d, \rho_k) = b^T d + \rho_k \max(\ell_c^k(d), 0)$$

and λ_k^j are the Lagrange multipliers associated with the solution to (18) and $\hat{P}_\infty(x, \rho)$ is an approximation to $P_\infty(x, \rho)$. This result is particularly insightful, since it shows that d will be approximately a direction of descent for $P_\infty(\cdot, \rho)$ at x_k. In particular, the following approximate directional derivative of $P_\infty(\cdot, \rho)$ at x_k in the direction d_k

$$v_k = \hat{P}_\infty(x_k + d_k, \rho_k) - P_\infty(x_k, \rho_k) \tag{20}$$

will be negative. The algorithm will take a serious step from x_k to $x_{k+1} = y_{k+1}$ if y_{k+1} is significantly better that x_k in the sense that

$$P_\infty(y_{k+1}, \rho_k) \leq P_\infty(x_k, \rho_k) + m v_k \tag{21}$$

where $m \in (0, 1)$ is a parameter. Otherwise, a null step $x_{k+1} = x_k$ will occur. In both cases the new subgradient $g_c^k = g_c(y_{k+1})$ information collected at y_{k+1} will be added to the bundle and will enable the method to generate a new search direction d_{k+1}.

Algorithm:

Step 0 (Initialization) Select a starting point x_1 and matrix B_1, a final accuracy $\varepsilon_f \geq 0$, a final infeasibility tolerance $\varepsilon_c \geq 0$ and a line search parameter $m \in (0, 1)$. Set $y_1 = x_1$, $J_c^1 = 1$, $g_c^1 = \nabla c(y_1)$ and $\rho_0 = 0$.

Step 1 (Direction Finding) Find the solution d_k to subproblem (18) and the corresponding Lagrange multipliers λ_k and compute $\tilde{\rho}_k = \sum_{j \in J_c^k} \lambda_k^j$

Step 2 (Penalty Updating) If $\tilde{\rho}_k \leq \rho_{k-1}$, set $\rho_k = \rho_{k-1}$; else set $\rho_k = max(\tilde{\rho}_k, 2\rho_{k-1})$

Step 3 (Stopping Criterion) Compute v_k by (20). If $v_k \geq -\varepsilon_f$ and $c(x_k) \leq \varepsilon_c$, terminate; otherwise, continue.

Step 4 (Line Search) Set $y_{k+1} = x_k + d_k$. If (21) holds, set $x_{k+1} = y_{k+1}$ (serious step); otherwise $x_{k+1} = x_k$ (null step).

Step 5 (Linearizations Updating) Select $J_c^{k+1} \supset J_c^k \cup \{k+1\}$. Set $g_c^{k+1} = \nabla c(y_{k+1})$ and

$$\begin{aligned} c_j^{k+1} &= c_j^k + \langle g_c^j, x_{k+1} - x_k \rangle \\ c_{k+1}^{k+1} &= c(y_{k+1}) + \langle g_c^{k+1}, x_{k+1} - y_{k+1} \rangle \end{aligned}$$

The algorithm given above is in its most basic form as presented in [10] and the reader is referred to [10] for proof of convergence of the above algorithm.

Remark 3.1 *It should be observed that* **Step 1** *is solved more efficiently by solving the dual problem associated with (18), since in the dual case the problem size depends on the size of the bundle of information which in general is much smaller than the length of d*

We have not until now discussed the choice of the matrix B^k. In [11] the matrix is chosen as $B_k = \mu_k I$ and μ_k has the function of controlling a *trust region* around the point x_k. In our limited numerical experimentation so far conducted, it was found that it was most efficient to update B^k using a Quasi Newton BFGS update:

$$\begin{aligned} s_k &= y_{k+1} - x_k \\ z_k &= g_c(y_{k+1}) - g_c(x_k) \\ B_{k+1} &= B_k - \frac{B_k s_k^T s_k B_k}{s_k^T B_k s_k} + \frac{z_k z_k^T}{z_k^T s_k} \end{aligned}$$

4 Numerical Results

To evaluate the performance of the method described in the previous sections, the two-mass/spring system from [15] is considered. The system has the following state-space form:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} w + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u \\ z &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \end{bmatrix} w + [0] u \\ y &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 1 \end{bmatrix} w \end{aligned}$$

The full order optimal $\mathcal{H}_\infty$ controller has an $\mathcal{H}_\infty$ norm of $\gamma = 1$. The objective is to find a second order ($n_c = 2$) controller (the plant cannot be stabilized with a controller of order one) with the smallest possible γ. In order to evaluate the performance of the proposed method, a comparison is made with the method found in [14]. This comparison is shown in Table 1 where in the first column is the the achieved $\mathcal{H}_\infty$ performance γ, in the second column is the number of (outer) iterations and the third column shows the number of function evaluations necessary for the specific method to converge. The comparison between the two methods is quite fair due to the following reasons: firstly, both methods use the same subroutine to compute the constraint function and it's gradient. Secondly, both methods solve a Quadratic Programming (QP) problem at each iteration. Finally, both methods are essentially programmed in Matlab. A comparison on the CPU time taken by both methods would be misleading since the present method makes use of QP solver which is already compiled. The same problem has also been

Method	$\mathcal{H}_\infty$ Perf.	Iter	Nf
Bundle	4.109	138	192
[14]	4.102	185	524

Table 1: Comparison of numerical methods for reduced order controllers for two mass system

solved in [15] and [8] in which the achieved values of γ were 4.96 and 4.14 respectively. In this particular case the bundle method did exceptionally well, especially in the number of function evaluations. This is particularly important because in this particular case function evaluations are expensive.

A second test example is a four disk control system studied by Enns and taken from [16]. The full 8^{th} order optimal $\mathcal{H}_\infty$ controller has an $\mathcal{H}_\infty$ norm of $\gamma = 1.1272$. Table 2 shows the results obtained applying the present method and the method from [14] to Enns problem. In [16] several weighted controller reduction methods are applied to Enns problem, the best $\mathcal{H}_\infty$ performance of all the methods for each controller order is also given in Table 2. In this example the bundle is still superior

Order	Method	Iter.	Nf	γ
$n_c = 6$	Bundle	651	1134	1.140
	Ref.[14]	566	3122	1.164
	Ref.[16]	-	-	1.196
$n_c = 5$	Bundle	643	1229	1.140
	Ref.[14]	594	2901	1.164
	Ref.[16]	-	-	1.195
$n_c = 4$	Bundle	779	1566	1.180
	Ref.[14]	552	1867	1.239
	Ref.[16]	-	-	1.195
$n_c = 3$	Bundle	742	1589	1.576
	Ref.[14]	620	2238	1.351
	Ref.[16]	-	-	1.488
$n_c = 2$	Bundle	769	1526	1.292
	Ref.[14]	551	2001	1.297
	Ref.[16]	-	-	1.417
$n_c = 1$	Bundle	971	1802	1.75
	Ref.[14]	596	2128	1.656
	Ref.[16]	-	-	2.467

Table 2: Comparison of numerical methods for reduced order controllers

from a number of function evaluations point of view. It does, however, typically involve more iterations with respect to [14].

Another important observation can be made from the data relative to the $\mathcal{H}_\infty$ performance γ. Except for $n_c = 3$ (we have not yet investigated why this happens) both methods always obtain a value of γ less than the ones obtained in [16].

5 Conclusions

In this paper we have presented a novel approach to computing fixed order controllers. The method makes use of the bundle method for nondifferentiable optimization. Our limited computational experience has shown that the method in its current from is more efficient than the method proposed in [14]. Future work will involve further optimization of the code and more numerical experiments.

References

[1] E. Beran and K.M. Grigoriadis. A combined alternating projections and semidefinite programming algorithm for low–order control design. In *Proc. 13th. Int. Fed. Auto. Contr.*, San Francisco, CA, 1996.

[2] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *Int. Journal of Robust and Nonlinear Control*, 4(4):421–448, July-August 1994.

[3] P. Gahinet and A. Ignay. Low–order H_∞ synthesis via LMIs. In *Proc. Amer. Contr. Conf.*, pages 1499–1500, Baltimore, MD, June 1994.

[4] L. El Ghaoui, F. Oustry, and M. Aitrami. A cone complimentarity linearization algorithm for static output-feedback and related problems. *IEEE Transactions on Automatic Control*, 42(8):1171–1176, 1997.

[5] G.H. Golub and C.F. van Loan. *Matrix Computations*. The John Hopkins University Press, third edition, 1996.

[6] K.M. Grigoriadis and R.E. Skelton. Low–order control design for LMI problems using alternating projections methods. *Automatica*, 32(8):1117–1125, 1996.

[7] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal of Optimization*, 10(3):673–696, 2000.

[8] J. Imae and T. Furudate. A design method for fixed-order H_∞ controllers via bilinear matrix inequalities. In *Proc. Amer. Contr. Conf.*, pages 1876–1880, San Diego, CA, June 1999.

[9] K.C. Kiwiel. *Methods of Descent for NonDifferentiable Optimization*, volume 1133 of *Springer Lecture Notes in Mathematics*. Springer-Verlag, Vienna, Austria, 1985.

[10] K.C. Kiwiel. A constraint linearization method for nondifferentiable convex minimization. *Numerische Mathematik*, 51:395–414, 1987.

[11] K.C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46(1):105–122, January 1990.

[12] S.A. Miller and R.S. Smith. A bundle method for efficiently solving large structured structured linear matrix inequalities. In *Proc. Amer. Contr. Conf.*, Chicago, IL, June 2000.

[13] S.A. Miller and R.S. Smith. Solving large structured semidefinite programs using an inexact spectral bundle method. In *Proc. IEEE Conf. Dec. Contr.*, volume 5, pages 5027–5032, Sydney, Australia, December 2000.

[14] M. Rotunno and R.A. de Callafon. Fixed order h_∞ control design for dual-stage hard disk drives. In *39th IEEE Conf. on Decision and Control*, Sydney, Australia, December 2000.

[15] R.E. Skelton, T. Iwasaki, and K.M. Grigoriadis. *A Unified Algebraic Approach to Linear Control Design*. Taylor and Francis, 1998.

[16] K. Zhou and J.C. Doyle. *Robust Control and Optimal Control*. Prentice-Hall, 1996.