2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

A Partially Augmented Lagrangian Method for Low Order H-infinity Controller Synthesis using Rational Constraints

Daniel Ankelhed, Anders Helmersson and Anders Hansson
Email: {ankelhed, andersh, hansson}@isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköping University, Sweden

***Abstract*— When designing robust controllers, H-infinity synthesis is a common tool to use. The controllers that result from these algorithms are typically of very high order, which complicates implementation. However, if a constraint on the maximum order of the controller is set, that is lower than the order of the (augmented) system, the problem becomes nonconvex and it is relatively hard to solve. These problems become very complex, even when the order of the system is low.**

The approach used in this work is based on formulating the constraint on the maximum order of the controller as a polynomial (or rational) equation. This equality constraint is added to the optimization problem of minimizing an upper bound on the H-infinity norm of the closed loop system subject to linear matrix inequality (LMI) constraints. The problem is then solved by reformulating it as a partially augmented Lagrangian problem where the equality constraint is put into the objective function, but where the LMIs are kept as constraints.

The proposed method is evaluated together with two well-known methods from the literature. The results indicate that the proposed method has comparable performance in most cases.

I. Introduction

The development of robust control theory emerged during the 80s and and a contributory factor certainly was the fact that the robustness of Linear Quadratic Gaussian (LQG) controllers can be arbitrarily bad as reported in [1]. A few years later, in [2], an important step in the development towards a robust control theory was taken, where the concept of H_∞ theory was introduced. The H_∞ synthesis, which is an important tool when solving robust control problems, was a cumbersome problem to solve until a technique was presented in [3], which is based on solving two Riccati equations. Using this method, the robust design tools became much easier to use and gained popularity. Quite soon thereafter, linear matrix inequalities (LMIs) were found to be a suitable tool for solving these kinds of problems by using reformulations of the Riccati equations, see [4].

Typical applications for robust control include systems that have high requirements for robustness to parameter variations and for disturbance rejection. The controllers that result from these algorithms are typically of very high order, which complicates implementation. However, if a constraint on the maximum order of the controller is set, that is lower than the order of the plant, the problem is no longer convex and it is then relatively hard to solve. These problems become very complex, even when the order of the system to be controlled is low. This motivates the development of efficient algorithms that can solve these kinds of problems.

In [5], Apkarian et. al presented a method for low order H_∞ controller synthesis which relaxes only one of the constraints and is thus called a *partially* augmented Lagrangian method. In [6] the method is extended to more general robust control than H_∞ controller problems and [7] generalizes the framework to optimization problems with general matrix inequality constraints.

In this paper we will describe a method based on what is done in [5], but where the equality constraint involves coefficients of a characteristic polynomial, similarly to what is done in some of our previous work, [8], [9]. One main difference compared to [8], [9] is that the method in this paper explicitly tries to minimize the performance measure γ instead of finding a controller that satisfies a pre-specified value. In contrast to the approach in [5], our method does not introduce additional variables when synthesizing dynamic controllers, i.e. controllers of order one or higher.

Other methods for solving reduced order H_∞ problems that have gained attention recently are e.g. HIFOO and HINFSTRUCT, see [10] and [11] respectively. These methods are based on nonconvex, nonsmooth approaches for minimizing the H_∞ norm of a closed loop system that do not involve any LMIs. The advantage of these methods is that they in general reduce the number of variables of the problem, while they on the other hand introduce other difficulties due to the nonsmooth formulation of the problem.

Denote with $\mathbb{S}^n$ the set of real symmetric $n \times n$ matrices and $\mathbb{R}^{m\times n}$ is the set of real $m \times n$ matrices. The notation $A \succ 0$ ($A \succeq 0$) and $A \prec 0$ ($A \preceq 0$) means A is a positive (semi)definite matrix and negative (semi)definite matrix, respectively.

II. Preliminaries

We begin by describing a linear system, G, with state vector, $x \in \mathbb{R}^{n_x}$. The input vector contains the disturbance signal, $w \in \mathbb{R}^{n_w}$, and the control signal, $u \in \mathbb{R}^{n_u}$. The output vector contains the measurement, $y \in \mathbb{R}^{n_y}$, and the performance measure, $z \in \mathbb{R}^{n_z}$. In terms of its system matrices, we can represent the linear system as

$$G:\left(\begin{array}{c}\dot{x}\\ \hline z\\ y\end{array}\right)=\left(\begin{array}{c|cc}A & B_1 & B_2\\ \hline C_1 & D_{11} & D_{12}\\ C_2 & D_{21} & D_{22}\end{array}\right)\left(\begin{array}{c}x\\ \hline w\\ u\end{array}\right), \quad (1)$$

where D_{22} is assumed to be zero, i.e., the system is strictly proper from u to y. If this is not the case, we can find a controller $\tilde{K}$ for the system where D_{22} is set to zero, and then construct the controller as $K = \tilde{K}(I + D_{22}\tilde{K})^{-1}$. Hence, there is no loss of generality in making this assumption. For simplicity, it is also assumed that the whole system is on minimal form, i.e., it is both observable and controllable. However, in order to find a controller, it is enough to assume detectability and stabilizability (non observable and non controllable modes are stable).

The linear controller is denoted K. It takes the system measurement, y, as input and the output vector is the control signal, u. The system matrices for the controller are defined by the equation

$$K:\begin{pmatrix}\dot{x}_K\\ u\end{pmatrix}=\begin{pmatrix}K_A & K_B\\ K_C & K_D\end{pmatrix}\begin{pmatrix}x_K\\ y\end{pmatrix}, \quad (2)$$

where $x_K \in \mathbb{R}^{n_k}$ is the state vector of the controller.

Lemma 1 (H_∞ controllers for continuous plants): The problem of finding a linear controller such that the closed loop system G_c is stable and such that $\|G_c\|_\infty < \gamma$, is solvable if and only if there exist positive definite matrices $X, Y \in \mathbb{S}^{n_x}$, which satisfy

$$\begin{pmatrix}\mathcal{N}_X & 0\\ 0 & I\end{pmatrix}^T\begin{pmatrix}XA+A^TX & XB_1 & C_1^T\\ B_1^TX & -\gamma I & D_{11}^T\\ C_1 & D_{11} & -\gamma I\end{pmatrix}\begin{pmatrix}\mathcal{N}_X & 0\\ 0 & I\end{pmatrix}\prec 0 \quad (3a)$$

$$\begin{pmatrix}\mathcal{N}_Y & 0\\ 0 & I\end{pmatrix}^T\begin{pmatrix}AY+YA^T & YC_1^T & B_1\\ C_1Y & -\gamma I & D_{11}\\ B_1^T & D_{11}^T & -\gamma I\end{pmatrix}\begin{pmatrix}\mathcal{N}_Y & 0\\ 0 & I\end{pmatrix}\prec 0 \quad (3b)$$

$$\begin{pmatrix}X & I\\ I & Y\end{pmatrix}\succeq 0 \quad (3c)$$

$$\operatorname{rank}(XY - I) \leq n_k. \quad (3d)$$

where $\mathcal{N}_X$ and $\mathcal{N}_Y$ denote any bases of the null-spaces of $\begin{pmatrix}C_2 & D_{21}\end{pmatrix}$ and $\begin{pmatrix}B_2^T & D_{12}^T\end{pmatrix}$ respectively.

Proof: See [4]. ■

It could be desirable to replace the rank constraint in (3d) with a smooth function in order to be able to apply gradient methods for optimization. To do this, the following lemma is used.

Lemma 2: Assume that the inequality

$$\begin{pmatrix}X & I\\ I & Y\end{pmatrix}\succeq 0 \quad (4)$$

holds. Let

$$\begin{aligned}\det(\lambda I-(I-XY)) &= \sum_{i=0}^{n_x} c_i(X,Y)\lambda^i =\\ &= \lambda^{n_x} + c_{n_x-1}(X,Y)\lambda^{n_x-1} + \ldots\\ &\quad + c_1(X,Y)\lambda + c_0(X,Y)\end{aligned} \quad (5)$$

be the characteristic polynomial of $(I - XY)$, where the functions $c_i(X,Y)$ are its coefficients. Then the following statements are equivalent if $n_k < n_x$:

1) $\operatorname{rank}(XY - I) \leq n_k$
2) $c_{n_x-n_k-1}(X,Y) = 0$

Additionally, all coefficients are non-negative, i.e.

$$c_i(X,Y) \geq 0,\ \forall i. \quad (6)$$

Proof: See [12]. ■

How to compute $c_i(X,Y)$ and their derivatives is explained in [12] where also additional properties of the coefficients are shown.

III. Problem formulation

The problem we wish to solve is this paper is to minimize γ subject to the constraints in (3). Formally this can be stated as the following optimization problem.

$$\begin{aligned}\text{minimize}\quad & \gamma\\ \text{subject to}\quad & c_{n_x-n_k-1}(X,Y) = 0\\ & (\gamma, X, Y) \in \mathbb{X}\end{aligned} \quad (7)$$

where $\mathbb{X}$ is a convex set defined by the three LMIs in (3a)–(3c). We have noticed that scaling the equality constraint function in (7) by the next coefficient in the characteristic polynomial in (5) makes it numerically sounder, i.e., we replace $c_{n_x-n_k-1}(X,Y)$ by $\hat{c}(X,Y) = c_{n_x-n_k-1}(X,Y)/c_{n_x-n_k}(X,Y)$. This results in the following problem.

$$\begin{aligned}\text{minimize}\quad & \gamma\\ \text{subject to}\quad & \hat{c}(X,Y) = 0\\ & (\gamma, X, Y) \in \mathbb{X}\end{aligned} \quad (8)$$

This problem can be solved by using the *partially augmented Lagrangian* algorithm, see e.g. [5], where the equality constraint is relaxed and added to the objective function in the following way.

$$\begin{aligned}\text{minimize}\quad & \gamma + \lambda\hat{c}(X,Y) + \frac{\mu}{2}\hat{c}^2(X,Y)\\ \text{subject to}\quad & (\gamma, X, Y) \in \mathbb{X}\end{aligned} \quad (9)$$

where λ is a Lagrangian multiplier and μ is a penalty multiplier. The word “partially” refers to the fact that only the equality constraint is used in the augmentation while the LMIs are kept as they are in order to keep the structure of the problem. The solution to the original problem (8) is obtained by iteratively solving an approximation of (9) for a sequence of increasing values of μ. More details on augmented Lagrangian methods can be found in e.g. the books by Bertsekas, [13], [14] and Nocedal and Wright, [15].

IV. Reformulating the problem

To simplify the notation, let us first define the half-vectorization operator.

Definition 1 (Half-vectorization): Let

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & & \vdots \\ \vdots & & \ddots & \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{pmatrix}.$$

Then

$$\text{vech}(X) = (x_{11}\ x_{21}\ \dots\ x_{n1}\ x_{22}\ \dots\ x_{n2}\ x_{33}\ \dots\ x_{nn})^T,$$

i.e., vech stacks the columns of X from the principal diagonal downwards in a column vector. See [16] for properties and details.

Next, let us do a variable substitution as follows. Let

$$x = \left(\text{vech}(X)^T, \text{vech}(Y)^T, \gamma\right)^T.$$

By choosing b as the last unit vector such that $\gamma = b^T x$, the optimization problem (9) can be written as

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \Phi_c(x, \lambda, \mu) \\ \text{subject to} \quad & x \in \mathbb{X} \end{aligned} \tag{10}$$

where

$$\Phi_c(x, \lambda, \mu) = b^T x + \lambda \hat{c}(x) + \frac{\mu}{2}\hat{c}^2(x).$$

This is a nonconvex problem, since the function $\hat{c}(x) = c_{n_x-n_k-1}(x)/c_{n_x-n_k}(x)$ is nonconvex. However $\mathbb{X}$ is a convex set which makes the problem somewhat less difficult to solve than a general nonconvex problem.

V. Calculating the search direction

The next step is to approximate $\Phi_c(x + p, \lambda, \mu)$ by a quadratic function related to the first three terms in the Taylor series expansion around the point x. Similarly to what is done in regular Newton methods, we intend to find a step direction p that minimizes this second order model, but the difference is that we also require that $x + p \in \mathbb{X}$, i.e. that the next point also lies in the feasible set. This problem can be formulated as

$$\begin{aligned} \underset{p}{\text{argmin}} \quad & \nabla_x \Phi_c(x, \lambda, \mu)^T p + \frac{1}{2} p^T H(x, \lambda, \mu) p \\ \text{subject to} \quad & x + p \in \mathbb{X} \end{aligned} \tag{11}$$

which is a conic programming problem that can be solved efficiently using e.g. Yalmip, [17] with SDPT3, [18]. The symmetric matrix $H(x, \lambda, \mu, \delta)$ is a positive definite approximation of the Hessian of $\Phi_c(x, \lambda, \mu)$. We will come back to how this approximation is calculated later.

A. Calculating the derivatives

In order to solve (11), we need to calculate the gradient and Hessian of Φ_c. Differentiating $\Phi_c(x, \lambda, \mu)$ with respect to x yields

$$\nabla_x \Phi_c(x, \lambda, \mu) = b + \lambda \nabla_x \hat{c}(x) + \mu \hat{c}(x) \nabla_x \hat{c}(x)$$
$$\nabla_{xx} \Phi_c(x, \lambda, \mu) = \left(\lambda + \mu \hat{c}(x)\right) \nabla^2_{xx} \hat{c}(x) + \mu \nabla_x \hat{c}(x) \nabla^T_x \hat{c}(x)$$

with

$$\begin{aligned} \nabla \hat{c} &= \frac{1}{c_{n_x-n_k}} \nabla c_{n_x-n_k-1} - \frac{c_{n_x-n_k-1}}{c^2_{n_x-n_k}} \nabla c_{n_x-n_k} \\ \nabla^2 \hat{c} &= \frac{1}{c_{n_x-n_k}} \nabla^2 c_{n_x-n_k-1} - \frac{c_{n_x-n_k-1}}{c^2_{n_x-n_k}} \nabla^2 c_{n_x-n_k} \\ &+ \frac{2c_{n_x-n_k-1}}{c^3_{n_x-n_k}} (\nabla c_{n_x-n_k} \nabla^T c_{n_x-n_k}) \\ &- \frac{1}{c^2_{n_x-n_k}} (\nabla c_{n_x-n_k-1} \nabla^T c_{n_x-n_k} \\ &+ \nabla c_{n_x-n_k} \nabla^T c_{n_x-n_k-1}) \end{aligned}$$

where we have omitted the dependence on x to simplify notation. Since the constraint function $\hat{c}(x)$ is nonconvex, the Hessian $\nabla^2_{xx}\hat{c}(x)$ is not always positive definite which in turn might lead to that $H(x, \lambda, \mu) = \nabla^2_{xx}\Phi_c(x, \lambda, \mu)$ is not necessarily positive definite, which has to be dealt with. Two common ways are to either use Newton methods in which the Hessian is convexified or to use Trust-region methods where the nonconvexity is dealt with by optimizing over a limited region in each iteration. The authors of [5] advice against using Trust-region methods since the complexity of such a method is too large in this case. Therefore, our choice is to convexify the Hessian $\nabla^2_{xx}\Phi_c(x, \lambda, \mu)$ as will be explained next.

B. Hessian modification

We have chosen to calculate the exact Hessian $\nabla^2_{xx}\Phi_c(x, \lambda, \mu)$, and then convexifying it using a modified indefinite symmetric factorization as described in [19]. The procedure is as follows. First calculate the indefinite symmetric factorization $\nabla^2_{xx}\Phi_c = P^T L D L^T P$, where L is lower triangular, P is a permutation matrix and D is a block diagonal matrix with block sizes of 1×1 or 2×2. Then we construct a modification matrix F such that $L(D + F)L^T$ is sufficiently positive definite. In order to calculate this modification matrix, first compute the eigenvalue factorization

$$D = Q\bar{D}Q^T. \tag{12}$$

Then calculate the modification matrix

$$F = QEQ^T,$$

where the diagonal matrix E is defined by

$$E_{ii} = \begin{cases} 0, & \text{if } \bar{D}_{ii} \geq \delta, \\ \delta - \bar{D}_{ii}, & \text{if } \bar{D}_{ii} < \delta, \end{cases} \quad i = 1, 2, \dots \tag{13}$$

The matrix F is now the minimal matrix in Frobenius norm such that $D + F \succeq \delta I$. Note that since D is tridiagonal, calculating the eigenvalue factorization in (12) is computationally cheap. The parameter δ is chosen as $10^{-4}\|\nabla^2_{xx}\Phi\|_\infty$, where the matrix norm $\|A\|_\infty$ denotes the largest row sum of A.

Now we are ready to outline the suggested algorithm for H_∞ synthesis.

VI. An outline of the algorithm

The algorithm can be outlined as follows.

1) **Initial phase.**
   a) Find a starting point by solving the convex SDP
   $$\begin{aligned} \underset{x}{\text{minimize}} \quad & \gamma + \operatorname{trace}(X+Y) \\ \text{subject to} \quad & (X,Y,\gamma) \in \mathbb{X} \end{aligned} \tag{14}$$
   and denote the solution $(X^{(0)}, Y^{(0)})$. The objective function in (14) is a combination of two objectives. The first objective is that the performance measure γ should be low and the second is that the equality constraint $\hat{c}(X,Y) = 0$ should be approximately satisfied. Minimizing $\operatorname{trace}(X+Y)$ is a heuristic for minimizing the rank of $I - XY$ that is used in e.g. [20], [21] and [22].
   b) Set $k := 0$. Choose starting values for $\lambda^{(0)}$ and $\mu^{(0)}$, the parameters $\rho > 1$ and $0 < \rho_0 < 1$ and the tolerance ϵ.
2) **Optimization phase.**
   Set $k := k+1$ and let $p_X, p_Y \in \mathbb{S}^{n_x}, p_\gamma \in \mathbb{R}$.
   a) Using $\lambda = \lambda^{(k-1)}$ and $\mu = \mu^{(k-1)}$), solve (11) for the solution
   $$p = \begin{pmatrix} \operatorname{vech}(p_X) \\ \operatorname{vech}(p_Y) \\ p_\gamma \end{pmatrix},$$
   which is the step direction.
   b) Update the variables as
   $$\begin{aligned} X^{(k)} &= X^{(k-1)} + \alpha p_X, \\ Y^{(k)} &= Y^{(k-1)} + \alpha p_Y, \\ \gamma^{(k)} &= \gamma^{(k-1)} + \alpha p_\gamma, \end{aligned}$$
   or equivalently
   $$x^{(k)} = x^{(k-1)} + \alpha p,$$
   where $\alpha = 0.98$.
3) **Update phase**.
   Update the Lagrangian multiplier λ using the following update rule.
   $$\lambda^{(k)} = \lambda^{(k-1)} + \mu^{(k-1)} \hat{c}(x^{(k)}) \tag{15}$$
   If $\hat{c}(x^{(k)}) > \epsilon$, update μ as follows.
   $$\mu^{(k)} = \begin{cases} \rho \mu^{(k-1)} & \text{if } \hat{c}(x^{(k)}) > \rho_0 \hat{c}(x^{(k-1)}) \\ \mu^{(k-1)} & \text{if } \hat{c}(x^{(k)}) \leq \rho_0 \hat{c}(x^{(k-1)}) \end{cases} \tag{16}$$
   The first option in (16) reflects our thought that the decrease in the equality constraint function value was not enough. Therefore we increase the penalty parameter. The second option reflects our content with the value of the constraint function, and we leave the penalty parameter at its current value.
4) **Terminating phase.**
   If $\hat{c}(x^{(k)}) > \epsilon$, go to phase 2, otherwise we check the following.
   - if $\gamma^{(k)} < 0.99\gamma^{(k-1)}$ for three consequent iterates, it is likely we are close enough to a local optimum. Proceed to phase 5.
   - Otherwise, the objective function value is still decreasing, hence we continue the optimization, i.e., go back to phase 2.
5) **Recover controller phase**.
   Recover the controller parameters (K_A, K_B, K_C, K_D) as described in [4] and verify that the closed loop system is stable and that $\|G_c\|_\infty < \gamma$ holds true. These requirements should normally be satisfied, but if there are numerical problems this might not hold true.

Remark 1: Note that in the optimization phase, one normally choose α in the interval $0 < \alpha \leq 1$ by performing a line search. However, we noticed that very small step-lengths α were taken which resulted in bad performance that might be caused by the *Maratos effect*. A solution could be to use a *watchdog* strategy to remedy this, but we have chosen to simply use $\alpha = 0.98$ which seem to work well. For more details on the Maratos effect and watchdog strategies, see [15].

VII. Numerical experiments

All experiments were performed on a Dell Optiplex GX620 with 2GB RAM, Intel P4 640 (3.2 GHz) CPU running under Windows XP using Matlab, version 7.11 (R2010b).

Evaluation of the methods was done on examples from the benchmark problem library COMPl$_{\text{e}}$ib, see [23]. The suggested method was evaluated and compared to Hifoo 3.0, see [10], and Hinfstruct which is based on the paper [11]. Hinfstruct is included in the Robust Control Toolbox in Matlab, version 7.11 (R2010b).

The results from the evaluation are presented in Table I, where the H_∞ norms and required computational times for the respective methods are displayed. Note that the same settings were used throughout the whole evaluation for the augmented Lagrangian method. Cases where the augmented Lagrangian method had numerical problems are marked by *. As suggested in [24] and [25] Hifoo was evaluated by running it ten times on each problem and choosing the best result. These results are displayed in Table I where the required time is summed over all ten runs. Hinfstruct was evaluated according to [26], by running it only once and initializing it with two extra starting points when comparing its performance with Hifoo, since Hifoo uses three randomized starting points. The effect of using additional randomized starting points for the augmented Lagrangian method has not been investigated.

The upper part of Table I shows the results from when controllers of either order zero or three were synthesized in order to evaluate both static output feedback controllers and reduced order feedback controllers. In cases where only the static output feedback controller is shown it is due to the fact that the higher order controllers turned out to have the same performance, thus there is no gain in using those results.

Since the computational complexity of HIFOO and HINFSTRUCT depend on the number of parameters in the controller while the augmented Lagrangian method does not, we chose to also include a system (IH) which has 11 input signals and 10 output signals in order to check if the results would differ. The number of decision variables for HINFSTRUCT and HIFOO is $n_k^2 + n_k n_y + n_u n_k + n_u n_y$ while for the augmented Lagrangian method it is $n_x(n_x+1)+1$, which means that the number of decision variables in our method is not affected by the number of states of the controller (n_k), inputs (n_u) or outputs (n_y), while HINFSTRUCT and HIFOO are. The results of this evaluation are shown in the lower part of Table I. For this example we also synthesized controllers of higher order than for the other examples. A comparison with the quasi-Newton method in [8] is presented in [27].

TABLE I

RESULTS FROM EVALUATION ON A COLLECTION OF SYSTEMS FROM COMPLeIB. THE FIRST COLUMN DISPLAYS THE SYSTEM NAME, THE ORDER OF THE SYSTEM, THE NUMBER OF INPUTS AND OUTPUTS AND THE ORDER OF THE CONTROLLER THAT WAS SYNTHESIZED. THE SECOND, THIRD AND FORTH COLUMNS SHOW THE H_∞ NORM AND REQUIRED TIME FOR THE AUGMENTED LAGRANGIAN METHOD (AL), HINFSTRUCT (HS) AND HIFOO (HF) RESPECTIVELY. CASES WHERE THE AUGMENTED LAGRANGIAN METHOD HAD NUMERICAL PROBLEMS ARE MARKED BY *. NOTE THAT THE REQUIRED TIME FOR HIFOO (t^{HF}) IS THE ACCUMULATED TIME FOR ALL TEN RUNS.

Sys, (n_x,n_u,n_y,n_k)	$\|\cdot\|_\infty^{\mathrm{AL}}$, t^{AL}	$\|\cdot\|_\infty^{\mathrm{HS}}$, t^{HS}	$\|\cdot\|_\infty^{\mathrm{HF}}$, t^{HF}
AC2 (5,3,3,0)	0.11, 19.1 s	0.11, 3.47 s	0.11, 168 s
AC5 (4,2,2,0)	670, 20.8 s	665, 1.80 s	669, 24.8 s
AC5 (4,2,2,3)	660*, 10.3 s	658, 3.88 s	643, 1100 s
AC18 (10,2,2,0)	14.8, 37.4 s	10.7, 2.97 s	12.6, 124 s
AC18 (10,2,2,3)	8.09, 36.9 s	6.51, 8.22 s	6.54, 3860 s
CM1 (20,1,2,0)	0.84, 278 s	0.82, 1.91 s	0.82, 125 s
EB4 (20,1,1,0)	2.46*, 460 s	2.06, 3.94 s	2.06, 10.5 s
EB4 (20,1,1,3)	2.14, 370 s	1.82, 7.78 s	1.82, 1160 s
JE3 (24,3,6,0)	8.74, 645 s	5.10, 5.31 s	5.10, 4880 s
JE3 (24,3,6,3)	2.89*, 1403 s	2.90, 11.6 s	2.89, 5910 s
IH (21,11,10,0)	1.88, 367 s	1.59, 38.0 s	1.90, 2450 s
IH (21,11,10,1)	1.86, 523 s	1.80, 43.0 s	1.80, 2410 s
IH (21,11,10,3)	1.49, 373 s	1.57, 51.0 s	1.74, 2170 s
IH (21,11,10,5)	1.39*, 868 s	1.15, 65.3 s	1.69, 2620 s
IH (21,11,10,7)	1.61*, 169 s	0.79, 86.2 s	1.72, 2450 s

VIII. RESULTS

The results in the upper part of Table I indicate that the augmented Lagrangian method achieves comparable results in most cases. HIFOO performs well but HINFSTRUCT obtains the best results overall and is by far the fastest algorithm. However it does not always find the best result of the three methods.

The results in the lower part of Table I show that even if the number of parameters in the controller are many, HINFSTRUCT achieves better results than the augmented Lagrangian method in all cases but one. For these problems HIFOO does not perform as well as for the problems in the upper part of the table and the required time is far more than required by the other methods. However, if time is an issue, either using fast mode or just running it once instead of ten times will reduce the required computational time.

IX. CONCLUSIONS AND FUTURE WORK

A. Conclusions

We have presented a method for low order H_∞ controller synthesis based on the LMI formulation of the problem. The approach is to reformulate the rank constraint as a rational equality constraint and then solve the problem by using a partial augmented Lagrangian minimization algorithm. The suggested method was evaluated and compared with two other methods from the literature. The evaluation indicates that the suggested algorithm obtains comparable results in most cases. Overall, HINFSTRUCT is the fastest of the three compared methods.

B. Future Work

We would like to improve the numerical properties of the method so that it becomes more stable and able to handle higher order systems. The impact of adding extra, possibly randomized, starting points would also be interesting to investigate.

X. ACKNOWLEDGMENTS

The author would like to thank The Swedish research council for financial support under contract no. 60519401.

REFERENCES

[1] J. Doyle, "Guaranteed margins for LQG regulators," *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 756–757, 1978.

[2] G. Zames, "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 301–320, Apr 1981.

[3] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–47, 1989.

[4] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to H_∞ control," *International Journal of Robust and Nonlinear Control*, vol. 4, no. 4, pp. 421–48, 1994.

[5] P. Apkarian, D. Noll, and H. Tuan, "Fixed-order H-infinity control design via a partially augmented Lagrangian method," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 12, pp. 1137–1148, 2003.

[6] B. Fares, P. Apkarian, and D. Noll, "An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory," *International Journal of Control*, vol. 74, no. 4, pp. 248–360, 2001.

[7] D. Noll, M. Torki, and P. Apkarian, "Partially augmented lagrangian method for matrix inequality constraints," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 161–184, 2004.

[8] D. Ankelhed, A. Helmersson, and A. Hansson, "A quasi-Newton interior point method for low order H-infinity controller synthesis," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1462–1467, June 2011.

[9] ——, "A primal-dual method for low order H-infinity controller synthesis," in *Proceedings of the 2009 IEEE Conference on Decision and Control*, Shanghai, China, Dec 2009.

[10] S. Gumussoy, D. Henrion, M. Millstone, and M. Overton, "Multiobjective robust control with HIFOO 2.0," in *Proceedings of the IFAC Symposium on Robust Control Design*, Haifa, Israel, June 2009, pp. 144–149.

[11] P. Apkarian and D. Noll, "Nonsmooth H_∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.

[12] A. Helmersson, "On polynomial coefficients and rank constraints," Department of Automatic Control, Linköping university, Sweden, Tech. Rep. LiTH-ISY-R-2878, 2009. [Online]. Available: http://www.control.isy.liu.se/publications/doc?id=2119

[13] D. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. New York, USA: Academic Press, Inc., 1982.

[14] ——, *Nonlinear programming*. Belmont, Massachusetts, USA: Athena Scientific, 1995.

[15] J. Nocedal and S. Wright, *Numerical optimization*, 2nd ed. Springer, 2006.

[16] H. Lütkepohl, *Handbook of matrices*. John Wiley & Sons, Ltd, 1996.

[17] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004, available from http://users.isy.liu.se/johanl/yalmip/.

[18] K. Toh, M. Todd, and R. Tütüncü, "SDPT3 — a Matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999.

[19] S. Cheng and N. Higham, "A modified cholesky algorithm based on a symmetric indefinite factorization," *SIAM Journal on Matrix Analysis and Applications*, vol. 19, no. 4, pp. 1097–1110, 1998.

[20] R. Orsi, U. Helmke, and J. Moore, "A Newton-like method for solving rank constrained linear matrix inequalities," *Automatica*, vol. 42, no. 11, pp. 1875–82, 2006.

[21] T. Iwasaki, "The dual iteration for fixed-order control," *IEEE Transactions on Automatic Control*, vol. 44, no. 4, pp. 783–788, April 1999.

[22] M. Fazel, H. Hindi, and S. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *Proceedings of the American Control Conference*, vol. 6, Viginia, June 2001, pp. 4734–4739.

[23] F. Leibfritz, "COMPl$_e$ib: COnstrained Matrix optimization Problem library," 2006, available from http://www.complib.de/.

[24] S. Gumussoy and M. Overton, "Fixed-order H^∞ controller design via HIFOO, a specialized nonsmooth optimization package," in *Proceedings of the 2008 American Control Conference*, Seattle, 2008, pp. 2750–2754.

[25] ——, "Timings for numerical experiments on benchmark examples for fixed order H-infinity controller design," 2008. [Online]. Available: http://www.cs.nyu.edu/overton/papers/pdffiles/acc08times.pdf

[26] P. Apkarian, "E-mail correspondance," 2011.

[27] D. Ankelhed, "On design of low order H-infinity controllers," Linköping Studies in Science and Technology. Dissertations. No. 1371, Linköping University, The Institute of Technology, May 2011. [Online]. Available: http://www.control.isy.liu.se/publications/doc?id=2424