Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

A Nonsmooth Optimization Approach to H_∞ Synthesis

Musa A. Mammadov and Robert Orsi

Abstract— **A numerical method for solving the H_∞ synthesis problem is presented. The problem is posed as an unconstrained, nonsmooth, nonconvex minimization problem. The optimization variables consist solely of the entries of the output feedback matrix. No additional variables, such as Lyapunov variables, need to be introduced. The optimization procedure uses a line search mechanism where the descent direction is defined by a recently introduced dynamical systems approach. Numerical results for various benchmark problems are included.**

I. INTRODUCTION

The H_∞ synthesis problem involves finding an output feedback control matrix K that minimizes the H_∞ norm of a certain transfer function, subject to the constraint that K is stabilizing. This is a challenging problem and even finding a stabilizing K can be difficult. Indeed, if the entries of K are restricted to lie in prescribed intervals, then finding a stabilizing K is an NP-hard problem [5].

Existing numerical methods for the H_∞ synthesis problem are often based on first reformulating the problem into one involving linear matrix inequalities (LMIs) and an additional nonconvex rank constraint or nonconvex equality constraint. Numerical methods for such reformulations of the problem include those based on linearization [8], [16], [19]; alternating projections [13], [12], [25]; augmented Lagrangian methods [9], [4], [24], [3]; and sequential semidefinite programming [10].

The H_∞ synthesis problem can also be reformulating into a problem involving bilinear matrix inequalities (BMIs). Numerical methods for such reformulations of the problem include [10], [21], [17] and [27]. See also the references therein.

A disadvantage of these approaches is that they require the introduction of Lyapunov variables. As the number of Lyapunov variables grows quadratically with the number of state variables, the total number of variables can be quite large and even problems of moderate size can lead to numerical difficulties [2].

In this paper the H_∞ synthesis problem is posed as an unconstrained, nonsmooth, nonconvex minimization problem. The optimization variables for this reformulation consist solely of the entries of the output feedback matrix K and no additional variables, such as Lyapunov variables, need to be introduced. The approach taken to solve this problem is based on using the recently developed global optimization algorithm presented in [22] and [23]. This optimization algorithm uses a line search mechanism where the descent direction is defined via a dynamical systems approach. It can be applied to a wide range of functions, requiring only function evaluations to work. In particular it does not require gradient (or gradient like) information and hence it is well suited to optimizing our reformulation of the H_∞ synthesis problem.

Similar approaches, that is, ones based on directly minimizing an appropriate nonsmooth function of K, are taken in [6] in addressing various problems of robust stabilization, and in [1] and [2] for the H_∞ synthesis problem. The cost function we use is different to the ones used in these other works, as is our underlying method of optimization.

In addition, in [6] when optimizing robust stability and in [1] and [2] when dealing with the H_∞ synthesis problem, a stabilizing solution is first sought by trying to solve some auxiliary problem and then optimization is performed locally about this solution. While an initial stabilizing solution can be utilized by our algorithm, it is not required.

The paper is structured as follows. In Section II we recall the H_∞ synthesis problem as well as a specialization of this problem, the robust stabilization problem. In Section III we reformulate these problems as unconstrained optimization problems in K. We also mention some of the issues involved in trying to solve such problems. Section IV outlines the optimization approach used. Numerical experiments for various H_∞ synthesis and robust stabilization problems are presented in Section V. The paper ends with some concluding remarks.

II. PROBLEM FORMULATIONS

A. The H_∞ Synthesis Problem

Recall the *static output feedback H_∞ synthesis problem.*

Problem 1: Given a linear time invariant (LTI) system

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^{m_2}$ is the control, $y \in \mathbb{R}^{p_2}$ is the measured output, $w \in \mathbb{R}^{m_1}$ is the external input and $z \in \mathbb{R}^{p_1}$ is the controlled output, find a static output feedback

$$u = Ky$$

such that the H_∞ norm of $T_{w,z}(s, K)$, the closed loop transfer function from w to z, is minimal over the set of K for which $A + B_2KC_2$ is stable. □

M. A. Mammadov is with the School of Information Technology & Mathematical Sciences, University of Ballarat, Australia. m.mammadov@ballarat.edu.au

R. Orsi is with the Research School of Information Sciences and Engineering, Australian National University, Canberra ACT 0200, Australia. robert.orsi@anu.edu.au

We note that, given a system (1) and a output feedback matrix K, the closed loop dynamics from w to z are given by

$$\begin{bmatrix} \dot{x} \\ z \end{bmatrix} = \begin{bmatrix} A + B_2KC_2 & B_1 + B_2KD_{21} \\ C_1 + D_{12}KC_2 & D_{11} + D_{12}KD_{21} \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix}.$$

As is well known, the *dynamic output feedback H_∞ synthesis problem* can be posed as a static output feedback problem for an augmented system. Indeed, for a given system (1), suppose we would like to find an order $k \leq n$ dynamic controller of the form

$$\begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix}.$$

Here $x_K \in \mathbb{R}^k$. Then the dynamic output feedback H_∞ synthesis problem is equivalent to Problem 1 with the following substitutions:

$$K \to \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}, \; A \to \begin{bmatrix} A & 0 \\ 0 & 0_k \end{bmatrix}, \; B_1 \to \begin{bmatrix} B_1 \\ 0 \end{bmatrix},$$
$$B_2 \to \begin{bmatrix} 0 & B_2 \\ I_k & 0 \end{bmatrix}, \; C_1 \to \begin{bmatrix} C_1 & 0 \end{bmatrix}, \; C_2 \to \begin{bmatrix} 0 & I_k \\ C_2 & 0 \end{bmatrix},$$
$$D_{12} \to \begin{bmatrix} 0 & D_{12} \end{bmatrix}, \; D_{21} \to \begin{bmatrix} 0 \\ D_{21} \end{bmatrix}.$$

0_k and I_k denote the $k \times k$ zero and identity matrices respectively. Note that K which was $m_2 \times p_2$ has been replaced by a matrix of dimension $(k + m_2) \times (k + p_2)$.

B. The Robust Stabilization Problem

Before introducing the robust stabilization problem, we present some preliminaries.

If X is a square matrix, let $\alpha(X)$ denote the maximum of the real parts of the eigenvalues of X,

$$\alpha(X) := \max_i \operatorname{Re}(\lambda_i(X)).$$

Of course, X is stable if and only if $\alpha(X) < 0$.

For $X \in \mathbb{C}^{n \times n}$, let $\beta(X)$ denote its *complex stability radius* [14],

$$\beta(X) := \min\{\|E\| \mid E \in \mathbb{C}^{n \times n}, \alpha(X + E) \geq 0\}.$$

Here $\|\cdot\|$ denotes the maximum singular value norm, $\|E\| = \sigma_{\max}(E)$. $\beta(X)$ is zero if and only if X is unstable. The complex stability radius of a stable matrix X determines how robust the stability of X is with respect to additive (complex) perturbations of X. For any X, $\beta(X)$ gives the distance to the unstable matrices.

The *robust stabilization problem* is the following.

Problem 2: Given a linear time invariant (LTI) system

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control, and $y \in \mathbb{R}^p$ the output, find a static output feedback control law

$$u = Ky$$

that maximizes the complex stability radius of the closed loop system matrix $A + BKC$:

$$\max_{K \in \mathbb{R}^{m \times p}} \beta(A + BKC).$$

□

Problem 2 is a special case of Problem 1, as we now show. Suppose X is a stable matrix with associated transfer function $H(s) := (sI - X)^{-1}$. Then $\beta(X)$ and the H_∞ norm of H are related by

$$\beta(X) = \|H(s)\|_\infty^{-1}.$$

As a result, Problem 2 is equivalent to minimizing the H_∞ norm of the transfer function $(sI - (A + BKC))^{-1}$ subject to the constraint that $A + BKC$ is stable. Given A, B and C as in Problem 2, taking the same A, $B_1 = I$, $B_2 = B$, $C_1 = I$, $C_2 = C$, $D_{11} = 0$, $D_{12} = 0$ and $D_{21} = 0$, it can be readily shown that Problem 1 reduces to Problem 2.

III. A Nonsmooth, Nonconvex Optimization Problem

Using the terminology of Problem 1, define

$$f(K) := \begin{cases} -\|T_{w,z}(s, K)\|_\infty^{-1}, & \text{if } \alpha(A + B_2KC_2) < 0, \\ \alpha(A + B_2KC_2), & \text{if } \alpha(A + B_2KC_2) \geq 0. \end{cases} \tag{2}$$

We try to solve Problem 1 by solving the following unconstrained minimization problem:

$$\min_{K \in \mathbb{R}^{m_2 \times p_2}} f(K).$$

Our motivation for choosing this particular objective function is as follows. The set of stabilizing K's is $\{K \mid \alpha(A + B_2KC_2) < 0\}$ and our aim is to minimize $\|T_{w,z}(s, K)\|_\infty$ over this set. Finding a K that minimizes $\|T_{w,z}(s, K)\|_\infty$ is the same as finding a K that minimizes $-\|T_{w,z}(s, K)\|_\infty^{-1}$. However, using $-\|T_{w,z}(s, K)\|_\infty^{-1}$ has the following advantage. Within the stabilizing set, $-\|T_{w,z}(s, K)\|_\infty^{-1}$ is negative and converges to zero if $\alpha(A + B_2KC_2)$ converges to zero. It follows that f is a continuous function of K, that is a globally defined extension of $-\|T_{w,z}(s, K)\|_\infty^{-1}$. (Note that $\|T_{w,z}(s, K)\|_\infty$ does not have a useful continuous extension as it becomes unbounded as K goes to the boundary of the set of stabilizing K's.) Furthermore, f penalizes non-stabilizing K's. This obviates the need to deal with stability and H_∞ norm minimization separately.

The fact that $\|T_{w,z}(s, K)\|_\infty$ can only be evaluated at stabilizing K's makes minimizing this quantity more difficult. Non-stabilizing K's provide a rather limited amount of information in regard to this objective function. The only information they do provide is the extent to which they are in fact non-stabilizing. This information is given by the quantity $\alpha(A + B_2KC_2)$, and has been incorporated into f.

The algorithm which will be used to minimize f, see Section IV, only needs to be able to evaluate f in order to work. There exist efficient numerical methods for calculating H_∞ norms (and hence for calculating f). We use the Matlab function `hinfnorm`.

We now make some observations regarding the robust stabilization problem. These observations will of course also necessarily tell us something about the more general H_∞ synthesis problem.

If the problem we are considering is actually a robust stabilization problem, i.e., a case of Problem 2, then in the definition of f, the term $-\|T_{w,z}(s,K)\|_\infty^{-1}$ is just $-\beta(A+BKC)$. Both α and $-\beta$ are nonsmooth and nonconvex, and β, but not α, is locally Lipschitz [6]. As noted in [6], lack of convexity means finding a global minimizer of $-\beta$ can be expected to be difficult and lack of smoothness means it is not possible to use standard local optimization methods such as steepest descent and Newton type methods. (Apparently applying such local optimization methods leads to problems at points where the gradient of β is discontinuous.)

Therefore we have a nonsmooth, nonconvex global optimization problem; quite a difficult problem. To our advantage we have not had to introduce Lyapunov variables and hence we have a problem formulation in many less variables than we would have otherwise. Here are some additional, particular aspects of the problem that are worth keeping in mind.

As already mentioned, just finding a stabilizing solution can be a challenge in itself. The set of stabilizing K's can be quite small. For example, for the Boeing 767 system considered in Section V, the following is a stabilizing solution,

$$K = \begin{bmatrix} -1.7319 & -2.1035e{-5} \\ 4.5059e{+1} & 2.1706e{-4} \end{bmatrix}.$$

Changing the $(1,2)$ entry of this K by plus or minus 10^{-5} makes the closed loop system unstable. As the feasible region can be quite localized, one would expect that finding such solutions, and moreover finding globally optimal solutions, would be quite difficult. A global search would have to search quite small regions. This may not be feasible. For example, the calculation of a function value can be fairly time consuming; in the Boeing 767 problem, which has 55 states, to calculate the value of β at a point takes approximately 0.35 seconds on a 3 GHz Pentium 4 machine.

As we have already indicated, for Problem 1, the function we are really interested in minimizing is not defined for all K's. (Problem 2 is similar in that, while β is defined everywhere, it is 0 for all non-stabilizing K's.) In 'ordinary' constrained optimization (see [26] and references therein), it is still possible to evaluate the objective function outside the feasible region. This may be extremely helpful for finding deep local minimizers inside the feasible region. For Problems 1 and 2 we do not have this advantage. In fact, the feasible region, the set of stabilizing K's, cannot even be usefully quantified.

Finally, it is worth mentioning that finding K that minimizes $\alpha(A+BKC)$ is quite different to finding K that minimizes $-\beta(A+BKC)$. In the first case, one seeks to find a K that causes solutions of the closed loop system to decay to zero as quickly as possible. (We are assuming there exist K for which the closed loop system is stable.) No regard is given to how robustly stable $A+BKC$ is with respect to perturbations. In the second case, one optimizes robust stability. While K must stabilize the system, no regard is given to how quickly solutions decay to zero. In other words, in terms of optimality, the behaviors of the functions $\alpha(A+BKC)$ and $-\beta(A+BKC)$ are quite different.

IV. A Global Optimization Algorithm

To minimize f, we will use the recently developed global optimization algorithm AGOP, which is presented in [22] and [23]. AGOP is designed for solving unconstrained continuous optimization problems. It uses a line search mechanism where the descent direction is defined via a dynamical systems approach.

AGOP can be applied to a wide range of functions, requiring only function evaluations to work. In particular it does not require gradient information and can be used to find minima of non-differentiable functions. Briefly, it works as follows. Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is the function to be minimized. The algorithm must first be given a set of points, say $\Omega = \{x_1, ..., x_q\} \subset \mathbb{R}^n$. A suitable choice for an initial set of points is the set of vertices of a box centered around $x = 0$. Suppose that $x_\star \in \Omega$ has the smallest cost of the points in Ω, that is, that $f(x_\star) \le f(x)$ for all $x \in \Omega$. The set Ω and the values of f at each of the points in Ω allow us to generate a dynamical system and this dynamical system determines a possible descent direction v at the point $x_\star$. (The details of this are rather involved and are not presented here. We refer the reader to [22] for further information.) An inexact line search along this direction provides a new point $\hat{x}_{q+1}$. A local search about $\hat{x}_{q+1}$ is then carried out. This is done using a *direct search* method called *local variation*. This is an efficient local optimization technique that does not explicitly use derivatives and can be applied to nonsmooth functions. (A good survey of direct search methods can be found in [18].) Letting x_{q+1} denote the optimal solution of this local search, the set Ω is augmented to include x_{q+1}. Starting with this updated Ω, the whole process can be repeated. The process is terminated when v is approximately 0 (or a prescribed bound on the number of iterations is reached). The solution returned is the current $x_\star$, that is, the point in Ω with the smallest cost. (If f is continuously differentiable then the solution will be a local minima.)

Note that the convex hull of the set of points in the initial Ω is roughly where AGOP looks for a solution. However, because line search segments are not constrained to lie in some prescribed region, during its operation the algorithm may add to Ω points that are not in the convex hull of the original Ω. As a result, the solution produced by the algorithm may not lie in the convex hull of the initial set of points.

In applying the algorithm to the problems in the next section, Ω is often taken to be the vertices of a box of the form

$$\{K \in \mathbb{R}^{m_2 \times p_2} \mid |K_{ij} - \bar{K}_{ij}| \le \rho \text{ for all } i,j\}. \quad (3)$$

The box center $\bar{K}$ is initially taken as $\bar{K} = 0$.

Given a reasonable choice for ρ, the solution from an initial set of vertices, let us denote it by $K_\star$, is often stabilizing and $\|T_{w,z}(s, K_\star)\|_\infty$ can be quite small as well. However, if desired, the user can try to find an even more optimal solution by re-running the algorithm with $\bar{K} = K_\star$ and using either the same value of ρ or a smaller value. This produces another solution which can itself be used as a new $\bar{K}$ and this process can be repeated as long as desired.

Our aim in this paper is to show that the methods presented here can be successfully used for finding deep optimal solutions to H_∞ synthesis problems. How best to choose successive boxes from which to define Ω, optimal stopping criteria, and other such questions, have not been considered here. They are interesting questions for future investigations.

V. NUMERICAL EXPERIMENTS

This section contains some numerical experiments for various problems from the literature. Considered are both robust stabilization problems and H_∞ synthesis problems.

All computational results were obtained using a 3 GHz Pentium 4 machine. Our algorithm was coded using Matlab 7.0.

A. Turbo-generator: Robust Stabilization

The first system considered is a turbo-generator model from [15] (system TG1 from the *COMPl$_e$ib* collection [20]). For this system, $n = 10$ and $m = p = 2$. The A matrix for this system is stable with $\beta(A) = 0.00767$. Our aim is to find K that maximizes $\beta(A + BKC)$.

With Ω given by the vertices of the box in (3) with $\bar{K} = 0$ and $\rho = 5$, the algorithm found the following solution

$$K = \begin{bmatrix} 0.43526 & 1.0001 \\ -0.095954 & -0.14290 \end{bmatrix}, \tag{4}$$

for which $\beta(A+BKC) = 0.0739$. This value is substantially better than $\beta(A)$. Total solution time was 9.3 seconds.

Next, the algorithm was re-run with $\bar{K}$ given by K in (4) and $\rho = 1$. This gave the following solution,

$$K = \begin{bmatrix} -1.0203 & -1.1188 \\ -0.11116 & -0.16429 \end{bmatrix},$$

for which $\beta(A + BKC) = 0.0780$. This solution is better than the first solution though the improvement is fairly modest. The time taken for this second step was 16 seconds.

Robust stabilization of the turbo-generator model is also considered in [6]. The solution given there is

$$K = \begin{bmatrix} -0.7763 & -0.7193 \\ -0.0935 & -0.1515 \end{bmatrix},$$

for which $\beta(A + BKC) = 0.0785$. This value is slightly better than our own value.

Taking a different set Ω, we were also able to find a K which produces the same stability radius. Indeed, taking $\bar{K}_{ij} = -0.5$ and $\rho = 0.5$, gives

$$K = \begin{bmatrix} -0.98379 & -1.0554 \\ -0.098679 & -0.15851 \end{bmatrix},$$

for which $\beta(A + BKC) = 0.0785$. Solution time was 30 seconds. This further highlights that different Ω's may lead to different solutions.

Other $\bar{K}$'s and ρ's were also tried however it was not possible to further significantly improve $\beta(A+BKC)$. The best K found was

$$K = \begin{bmatrix} -0.86223 & -0.85477 \\ -0.093992 & -0.15384 \end{bmatrix},$$

for which $\beta(A+BKC) = 0.0786$. It is most likely that the global optimal value for this problem is 0.0786 or close to it.

B. Boeing 767: Robust Stabilization

The next system considered is a model of a Boeing 767 aircraft at a flutter condition [7] (system AC10 from the *COMPl$_e$ib* collection [20]). For this system, $n = 55$ and $m = p = 2$. The A matrix is unstable. In this subsection we consider for this system the problems of robust stabilization via static control and robust stabilization via low-order dynamic control.

Numerical methods capable of finding stabilizing controllers for the system have only recently appeared; see [6], [1] and [2]. Applying the algorithm with $\bar{K} = 0$ and various choices for ρ, we were initially unsuccessful in finding a stabilizing solution. Examining the system matrices reveals that, while the nonzero entries in B are of the same magnitude, the entries in the first row of C are roughly 10^5 times smaller in magnitude than the entries of the second row of C. That is, the problem is poorly scaled. This issue can be overcome by multiplying the second row of C by 10^{-5}. If a controller K could be found for this re-scaled system, a controller for the original unscaled system would be K with its last column multiplied by 10^{-5}. Using this re-scaling method, with $\bar{K} = 0$ and $\rho = 10$, the following static controller was found,

$$K = \begin{bmatrix} 2.3884 & 5.6913e{-7} \\ 5.2871 & 2.7660e{-5} \end{bmatrix}, \tag{5}$$

for which $\beta(A + BKC) = 6.81 \times 10^{-5}$. Time taken was 210 seconds.

In [6], robust stabilization of the system is considered for $k = 0$ (the static controller case), and $k = 1$ and $k = 2$ (low-order dynamic control). We now demonstrate that our algorithm is able to improve on the solutions in [6], which up to now have been the best available. To do this, for each k, we use the existing solution in [6] to help us define an appropriate Ω. If we denote the solution in [6] by $\tilde{K}$, Ω is taken to be the vertices of a box roughly centered at $\tilde{K}$. For each box, edge lengths are not necessarily equal and are taken to be roughly proportional to the magnitudes of the corresponding entries in $\tilde{K}$. The algorithm is run with this Ω and then possibly re-run, one or more times, using a refined Ω. The results below highlight that the algorithm is capable of improving on existing results though the process we have mentioned requires human intervention and hence is not currently fully automated. The results achieved are as follows.

$k = 0$. The solution obtained in [6] produces a stability radius of $\beta(A + BKC) = 7.91 \times 10^{-5}$. Note that this value is greater than the value produced by K in (5). However, the best solution found by our algorithm was

$$K = \begin{bmatrix} -1.3249 & -1.4779e{-5} \\ 4.2684e{+1} & 1.6041e{-4} \end{bmatrix}, \qquad (6)$$

for which $\beta(A + BKC) = 9.23 \times 10^{-5}$. This value is greater than the value from [6].

$k = 1$. The solution obtained in [6] produces a stability radius of 9.98×10^{-5}. The best solution found by our algorithm was

$$\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} = \begin{bmatrix} -1.0530e{-1} & -5.0163e{+1} & -2.5015e{-3} \\ -6.0702e{-5} & 1.6448 & 6.8933e{-6} \\ -7.6961e{-1} & 2.6326 & 1.1263e{-4} \end{bmatrix},$$

for which the stability radius is 2.00×10^{-4}. This is a significantly improved value.

$k = 2$. The solution obtained in [6] produces a stability radius of 1.02×10^{-4}. The best solution found by our algorithm was

$$\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} = \begin{bmatrix} -2.5278e{-2} & -1.5700e{-1} & -5.6533e{+1} & -3.2344e{-3} \\ 2.7429e{-1} & -1.4034 & 2.4156e{+1} & -1.8510e{-3} \\ 5.9569e{-5} & -3.3545e{-2} & 1.6173 & 1.5836e{-5} \\ -4.1751e{-1} & -2.3468e{-1} & 3.5466 & 1.8787e{-4} \end{bmatrix},$$

for which the stability radius is 2.22×10^{-4}. This stability radius is again much better than the value given in [6].

C. Boeing 767: H_∞ Synthesis

In this subsection we again consider the Boeing 767 system but this time consider the problem of H_∞ synthesis. The system re-scaling technique used in the prior subsection to calculate (5) is again employed. (As we are considering the H_∞ problem, we would normally have to scale the last row of D_{21} by the same factor used to scale the last row of C. For this problem, however, $D_{21} = 0$.)

$k = 0$. Taking $\bar{K} = 0$ and $\rho = 10$, the algorithm found the following solution,

$$K = \begin{bmatrix} -8.9569e{-1} & 1.8405e{-5} \\ 4.1215 & 4.3066e{-5} \end{bmatrix},$$

for which $\|T_{w,z}(s, K)\|_\infty = 13.4$. Total solution time was 260 seconds. For comparison purposes, the best result from the literature, see [2], has a H_∞ norm equal to 13.1.

$k = 1$. Taking $\bar{K} = 0$ and $\rho = 5$, the algorithm found the following solution,

$$\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} = \begin{bmatrix} -1.3893 & 0.15545 & -8.6745e{-5} \\ 8.2769 & 0.22304 & 1.3279e{-5} \\ 2.7554 & -0.34522 & 4.9694e{-5} \end{bmatrix},$$

for which $\|T_{w,z}(s, [A_K\ B_K;\ C_K\ D_K])\|_\infty = 10.2$. Total solution time was 610 seconds. The best result from the literature, again see [2], has the same H_∞ norm.

D. Transport Airplane: H_∞ Synthesis

The final system considered is a transport airplane [11] (system AC8 from the *COMPl$_e$ib* collection [20]). For this system, $n = 9$, $m_2 = 1$ and $p_2 = 5$. The A matrix is unstable.

Taking $\bar{K} = 0$ and $\rho = 5$, the algorithm found the following static controller,

$$K = \begin{bmatrix} 1.0156 & -1.0300 & -1.5001 & 0.074096 & 1.5314 \end{bmatrix},$$

for which $\|T_{w,z}(s, K)\|_\infty = 2.01$. Total solution time was 18 seconds. In [2], the result for this problem has the same H_∞ norm.

VI. Conclusions

In this paper the H_∞ synthesis problem was posed as an unconstrained, nonsmooth, nonconvex minimization problem in the entries of the output feedback matrix K. A numerical method for solving this reformulation of the problem was presented and application of the algorithm to various benchmark problems produced quite positive results. In particular, the algorithm was able to significantly improve on the best results appearing in the literature for robust stabilization of the Boeing 767 model. While these preliminary results are very promising, the algorithm is currently not fully automated and more work is needed in this regard.

In addition, in the future, rather than using the Matlab function `hinfnorm` for calculating H_∞ norms, a much faster Fortran routine will be used. Aside from having to do numerous H_∞ norm calculations, our algorithm does not require much additional computational effort and hence it is expected that this change will substantially reduce solution times.

VII. Acknowledgements

The second author acknowledges the support of the Australian Research Council through grant DP0450539.

References

[1] P. Apkarian and D. Noll, "Controller design via nonsmooth multi-directional search," submitted.

[2] ——, "Nonsmooth H_∞ synthesis," submitted.

[3] P. Apkarian, D. Noll, J.-B. Thevenet, and H. D. Tuan, "A spectral quadratic-SDP method with applications to fixed-order H_2 and H_∞ synthesis," in *Proceedings Asian Control Conf.*, Melbourne, Australia, 2004.

[4] P. Apkarian, D. Noll, and H. D. Tuan, "Fixed-order H_∞ control design via a partially augmented Lagrangian method," *Int. J. Robust Nonlinear Control*, vol. 13, pp. 1137–1148, 2003.

[5] V. Blondel and J. H. Tsitsiklis, "NP-hardness of some linear control design problems," *SIAM J. Control Optim.*, vol. 35, no. 6, pp. 2118–2127, 1997.

[6] J. V. Burke, A. S. Lewis, and M. L. Overton, "A nonsmooth, nonconvex optimization approach to robust stabilization by static output feedback and low-order controllers," in *Proceedings ROCOND*, Milan, 2003.

[7] E. J. Davison, *Benchmark problems for control system design: Report of the IFAC Theory Committee*. Laxenberg: IFAC, 1990.

[8] L. El Ghaoui, F. Oustry, and M. Ait Rami, "A cone complementarity linearization algorithm for static output-feedback and related problem," *IEEE Trans. on Automatic Control*, vol. 42, no. 8, pp. 1171–1176, 1997.

[9] B. Fares, P. Apkarian, and D. Noll, "An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory," *Int. J. Control*, vol. 74, no. 4, pp. 348–360, 2001.

[10] B. Fares, D. Noll, and P. Apkarian, "Robust control via sequential semidefinite programming," *SIAM J. Control Optim.*, vol. 40, no. 6, pp. 1791–1820, 2002.

[11] D. Gangsaas, K. R. Bruce, J. D. Blight, and U.-L. Ly, "Application of modern synthesis to aircraft control: Three case studies," *IEEE Trans. on Automatic Control*, vol. 31, pp. 995–1014, 1986.

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

[13] K. M. Grigoriadis and R. Skelton, "Low-order control design for LMI problems using alternating projection methods," *Automatica*, vol. 32, no. 8, pp. 1117–1125, 1996.

[14] D. Hinrichsen and A. J. Pritchard, "Stability radii of linear systems," *Systems and Control Letters*, vol. 7, pp. 1–10, 1986.

[15] Y. S. Hung and A. G. J. MacFarlane, *Multivariable feedback: A quasi-classical approach*, ser. Lecture Notes in Control and Information Sciences. Berlin, Heidelberg, New York: Springer, 1982.

[16] S. Ibaraki and M. Tomizuka, "Rank minimization approach for solving BMI problems with random search," in *Proceedings American Control Conference*, 2001, pp. 25–27.

[17] S. Kanev, C. Scherer, M. Verhaegen, and B. De Schutter, "Robust output-feedback controller design via a local BMI optimization," *Automatica*, vol. 40, no. 7, pp. 1115–1127, 2004.

[18] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM Review*, vol. 45, no. 3, pp. 385–482, 2003.

[19] F. Leibfritz, "An LMI-based algorithm for designing suboptimal static $\mathcal{H}_2/\mathcal{H}_\infty$ output feedback controllers," *SIAM J. Control Optim.*, vol. 39, no. 6, pp. 1711–1735, 2001.

[20] ——, "COMPl$_e$ib: COstrained Matrix-optimization Problem library - a collection of test examples for nonlinear semidefinite programs, control system design and related problems," University of Trier, Tech. Rep., 2003.

[21] F. Leibfritz and E. M. E. Mostafa, "An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems," *SIAM J. Optim.*, vol. 12, no. 4, pp. 1048–1074, 2002.

[22] M. A. Mammadov, "A new global optimization algorithm based on a dynamical systems approach," in *Proc. 6th International Conference on Optimization: Techniques and Applications*, Ballarat, Australia, 2004, also in: *Research Report 04/04, University of Ballarat, 2004.* http://www.ballarat.edu.au/ard/itms/publications/researchPapers/Papers_2004/04-04.pdf.

[23] M. A. Mammadov, A. M. Rubinov, and J. Yearwood, "Dynamical systems described by relational elasticities with applications to global optimisation," in *Continuous Optimization: Current Trends and Applications*, V. Jeyakumar and A. Rubinov, Eds. Springer, 2005.

[24] D. Noll, M. Torki, and P. Apkarian, "Partially augmented Lagrangian method for matrix inequality constraints," *SIAM J. Optim.*, vol. 15, no. 1, pp. 161–184, 2004.

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

[26] A. M. Rubinov and X. Q. Yang, *Lagrange-type functions in constrained non-convex optimization*, ser. Applied Optimization. Kluwer Academic Publishers, 2003, vol. 85.

[27] J.-B. Thevenet, D. Noll, and P. Apkarian, "Nonlinear spectral SDP method for BMI-constrained problems: Applications to control design," in *Proceedings ICINCO*, Portugal, 2004.