Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Constant D-scale μ-Synthesis Approach based on Nonsmooth Optimization

Emmanuel Prempain * **Ian Postlethwaite** *

* *Engineering Department, University of Leicester, University Road, Leicester LE1 7RH, U. K. (e-mail: ep26@le.ac.uk)*

Abstract: This paper presents a nonsmooth optimization technique for solving a special μ-synthesis control problem. Attention is focused on controller synthesis problems that involve real diagonal scalings. An academic example illustrates the synthesis algorithm and a comparison is made with the well-known DK-iteration algorithm. This paper shows that nonsmooth optimization synthesis can provide better solutions than the standard DK-iteration algorithm.

1. INTRODUCTION

Over the past few years, the application of nonsmooth, nonconvex optimization to non standard linear feedback synthesis has been a subject of interesting research e.g. Apkarian and Noll [2006a], Henrion and Overton [2006]. Many non standard control problems, such as, fixed-order controller synthesis and decentralized control, are by their nature nonconvex and hence are difficult to solve by standard means (e.g. Bernstein [1992]). Semi-definite relaxation conditions for those problems can sometimes be derived at the expense of some conservatism.

Recently, nonsmooth optimization algorithms have been developed to cope with fixed-order feedback synthesis, decentralized control and other related polynomial and matrix optimization problems (Henrion and Overton [2006], Burke et al. [2006], Apkarian and Noll [2006a]). More sophisticated algorithms, combining multidirectional search with nonsmooth optimization techniques, are given in (Apkarian and Noll [2007]). Multidirectional search and nonsmooth optimization strategies are utilized in (Apkarian and Noll [2007]) for solving fixed-order output feedback controller synthesis, simultaneous stabilization and multi-objective controller synthesis problems. In Apkarian and Noll [2006b] local nonsmooth optimization algorithms are presented to allows one to synthesize controllers under integral quadratic constraints.

This paper focuses on a special scaled $\mathcal{H}_\infty$ controller synthesis. The problem consists of determining the parameters of an output feedback controller together with those of a real similarity scaling matrix such that the $\mathcal{H}_\infty$ norm of the (scaled) closed-loop transfer matrix is minimized. This problem is standard and appears in robust and gain scheduled control theories (Packard and Doyle [1993], Apkarian and Gahinet [1995], Apkarian and Adams [1998]). This problem is known to be nonconvex and a common way to solve it (approximately) is to use the well-known DK-iteration procedure (e.g. Balas et al. [2001]). The aim of this paper is to give an alternative solution based on a nonsmooth optimization.

The μ synthesis can be reformulated as finding the local minima of the unconstrained optimization program:

$$\text{minimize} \quad f(X), \quad X \in \mathbf{R}^N,$$

where f is a continuously differentiable function of X, and where the gradient of $f(X)$ can be easily computed. In this paper, f will represent the $\mathcal{H}_\infty$ norm of a scaled closed-loop transfer matrix which depends both on the controller and the scaling parameters. X is the vector of decision variables including the controller and scaling parameters to be optimized.

The main motivation of this paper is to show that some μ-synthesis problems, like the one of this paper, can be easily solved by taking advantage of the freely available hybrid nonsmooth optimization Matlab solver HANSO Overton [2006]. This nonsmooth unconstrained optimization solver combines gradient sampling and bundle methods and aims to find the local minima of a continuously differentiable function if its gradient can be computed almost everywhere.

The paper is structured as follows. Section 2 describes the control synthesis problem. Section 3 outlines the synthesis algorithm. In Section 4, the effectiveness of our μ-synthesis algorithm is illustrated on a satellite example taken from the μ-Analysis and Synthesis toolbox Balas et al. [2001] and a comparison with the standard μ-synthesis algorithm is given. Conclusions are given in section 5.

The notation is standard. If P and K are Linear Time-Invariant (LTI) systems, the notation $\mathcal{F}(P, K)$ denotes the closed-loop system resulting from the interconnection of systems P and K. T_{zw} is the transfer matrix relating the signal output z to the input signal w. $\|G(s)\|_\infty$ is the $\mathcal{H}_\infty$ norm of the LTI system $G(s)$, which for continuous-time systems, is given by

$$\|G(s)\|_\infty = \sup_{\omega \in \mathbf{R}} \bar{\sigma}(G(j\omega)),$$

where $\bar{\sigma}(M)$ is the largest singular value of matrix M. I stands for the identity matrix and I_n is the identity matrix of dimension n. M^* is the conjugate transpose of the matrix M and for a non singular matrix M, M^{-*} stands for $(M^{-1})^*$. If $f(X)$ is a complex or a real function of a matrix X then the derivative of $f(X)$ with respect to X is defined as $\frac{\partial}{\partial X} f(X) = \left[\frac{\partial}{\partial x_{ij}} f(X)\right]$.

2. PROBLEM STATEMENT

We consider the n^{th} order linear time-invariant generalized plant $P(s)$ with state-space equations:

$$\dot{x} = Ax + B_w w + B_u u, \tag{1}$$
$$z = C_z x + D_{zw} w + D_{zu} u, \tag{2}$$
$$y = C_y x + D_{yw} w + D_{yu} u, \tag{3}$$

where $x \in \mathbf{R}^n$ is the plant state vector, $w \in \mathbf{R}^{n_w}$ is partitioned as $[p^T, \tilde{w}^T]^T$, where $\tilde{w}$ includes disturbances, measurement noise, reference signals etc and where $p \in \mathbf{R}^{n_p}$ is the output signal from a block-diagonal uncertainty Δ. $z^T = [q^T, \tilde{z}^T]^T \in R^{n_z}$, where $\tilde{z}$ is the error signal and $q \in \mathbf{R}^{n_q}$ is the input signal to the block-diagonal uncertainty Δ. $u \in \mathbf{R}^{n_u}$ is the control input and $y \in \mathbf{R}^{n_y}$ is the measured input. The signals p and q satisfy $p = \Delta q$ where Δ belong to the set $\underline{\Delta}$ defined as

$$\underline{\Delta} = \{\text{blockdiag}(\theta_1 I_{r_1}, \ldots, \theta_m I_{r_m}) : \theta_i \in \mathbf{R}\}.$$

System (1)-(3) is supposed to be stabilizable and detectable. This assumption ensures the existence of an internally stabilizing output feedback control law $u = K(s)y$ of order n. Also we will suppose $D_{yu} = 0$. This assumption makes the closed-loop state-space matrices linear in the control matrices. This is a standard assumption that can always be satisfied via loop loop transformation (see e.g. Zhou et al. [1995] for details). We suppose, without loss of generality, that $n_w = n_z$ and we define integers $r > 0$ and $s > 0$ as $r := n_p = n_q$ and $s := n_w - r$.

The similarity matrices $J \in R^{r\times r}$ associated with $\underline{\Delta}$ are the real matrices J which commute with Δ, that is, the matrices J in

$$J_{\underline{\Delta}} := \{J \in R^{r\times r} \mid J\Delta = \Delta J, \quad \Delta \in \underline{\Delta}\}.$$

Definition 1. Scaled $\mathcal{H}_\infty$ synthesis The γ-suboptimal scaled $\mathcal{H}_\infty$ synthesis problem consists of finding an internally stabilizing output control law $u = K(s)y$ and a scaling matrix $J \in J_{\underline{\Delta}}$ such that

$$\left\| L\mathcal{F}(P(s), K(s))L^{-1} \right\|_\infty < \gamma, \tag{4}$$

where $L = \begin{pmatrix} J & 0 \\ 0 & I_s \end{pmatrix}$.

Inequality (4) implies that the loop of figure 1 is well-posed, internally stable, and $\|T_{\tilde{z}\tilde{w}}\|_\infty < 1/\gamma$ for all $\Delta \in \underline{\Delta}$ with $\|\Delta\|_\infty < 1/\gamma$ where $T_{\tilde{z}\tilde{w}}$ is the closed-loop transfer matrix relating the error signal $\tilde{z}$ to the exogenous signal $\tilde{w}$ (figure 1).

The above $\mathcal{H}_\infty$ controller synthesis problem, which is based on a scaled version of the small gain theorem, is a well-known nonconvex problem; see e.g. Apkarian and Gahinet [1995]. A standard way to deal with such a problem consists of using the so-called DK-iteration algorithm used in μ-synthesis Packard and Doyle [1993], Balas et al. [2001]. The DK-iteration algorithm involves a sequence of convex minimizations, first over the regulator (holding the scaling variable J fixed), and then over the scaling J for a fixed value of the regulator. However the DK-iteration procedure is not guaranteed to converge to the minimum μ value, even if it often works well in practice.

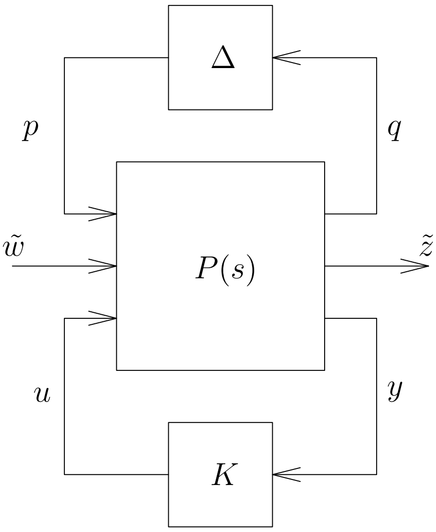

Fig. 1. Robust output feedback setting

Given a realization

$$K(s) = C_K(sI - A_K)^{-1}B_K + D_K, \quad A_K \in \mathbf{R}^{n\times n},$$

of the output controller, a realization of the scaled closed-loop transfer function T_{zw}, where

$$T_{zw}(s) := L\mathcal{F}(P(s), K(s))L^{-1}, \tag{5}$$

is

$$A_{cl} = \begin{pmatrix} A + B_u D_K C_y & B_u C_K \\ B_K C_y & A_K \end{pmatrix}, \tag{6}$$
$$B_{cl} = \begin{pmatrix} B_w + B_u D_K D_{yw} \\ B_K D_{zw} \end{pmatrix} L^{-1}, \tag{7}$$
$$C_{cl} = L\,(\, C_z + D_{zw} D_K C_y \;\; D_{zu} C_K\,)\,, \tag{8}$$
$$D_{cl} = L(D_{zw} + D_{zu} D_K D_{yw})L^{-1}. \tag{9}$$

Define the function

$$f = \sup_{w\in\mathbf{R}} \overline{\sigma}(T_{zw}(j\omega)), \tag{10}$$

f is just the $\mathcal{H}_\infty$ norm of the scaled closed-loop transfer matrix T_{zw}. It can be shown that f is continuous and continuously differentiable almost everywhere.

The derivatives of f with respect to the closed-loop state-space matrices and the scaling matrix L are given by:

$$\frac{\partial f}{\partial A_{cl}} = \phi^* C_{cl}^* u v^* B_{cl}^* \phi^* \tag{11}$$
$$\frac{\partial f}{\partial B_{cl}} = \phi^* C_{cl}^* u v^* L^{-*} \tag{12}$$
$$\frac{\partial f}{\partial C_{cl}} = L^* u v^* B_{cl}^* \phi^* \tag{13}$$
$$\frac{\partial f}{\partial D_{cl}} = L^* u v^* L^{-*} \tag{14}$$
$$\frac{\partial f}{\partial L} = uv^* L^{-*} M^* - L^{-*} M^* L^* u v^* L^{-*} \tag{15}$$

where $\phi = (j\omega I - A_{cl})^{-1}$, $M = L^{-1}(C_{cl}\phi B_{cl} + D_{cl})L$ and where u and v are respectively the left and the right singular vectors corresponding to the largest singular value of the scaled closed-loop transfer matrix T_{zw} across frequency.

If A, B, X and C are matrices of compatible dimensions, it can be shown that the derivative of $f(A + BXC)$ with respect to the matrix X is B^*GC^* where $G = \left[\frac{\partial}{\partial x_{ij}} f(X)\right]$. Because the controller matrices enter affinely into the closed-loop state-space matrices, the gradient of f with respect to the controller matrices can be easily derived from (11)-(14) using rule given above.

3. ALGORITHM

The minimization of f is done with the Matlab package HANSO which locally optimizes nonconvex, nonsmooth functions. HANSO requires the gradient of the function to be optimized, which, in this case, can be deduced from (11)-(15). It is worth recalling that HANSO does not require f or its gradient to be defined everywhere. For more details on the algorithms used by HANSO the reader is refereed to Overton [2006].

As is the case for any local optimization solver, HANSO requires an initial starting point. Here, a sensible initialization consists of using the parameters of an optimal full-order $\mathcal{H}_\infty$ controller obtained with L equal to the identity matrix.

4. EXAMPLE

The example is taken from the μ-Analysis and Synthesis Matlab toolbox Balas et al. [2001].

The state-space equations of a spinning satellite model G are given by

$$\begin{aligned}\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} &= \begin{pmatrix}0 & 10\\ -10 & 0\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}\\ &+ \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}u_1\\ u_2\end{pmatrix}\\ \begin{pmatrix}y_1\\ y_2\end{pmatrix} &= \begin{pmatrix}1 & 10\\ -10 & 1\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}\end{aligned}$$

The model of the first channel actuator uncertainty is represented by a multiplicative uncertainty with frequency weight

$$W_{11} = \frac{10s + 40}{s + 200}.$$

The model of the second channel actuator uncertainty is represented by a multiplicative uncertainty with frequency weight

$$W_{12} = \frac{10s + 240}{3s + 600}.$$

The total actuator uncertainty model is then $W_1 = diag(W_{11}, W_{12})$. The open-loop interconnection structure has uncertainty in each input channel. The shape of the frequency content of the disturbances acting at the plant output is given by

$$W_p = \frac{s + 4}{2s + 0.04} I_2.$$

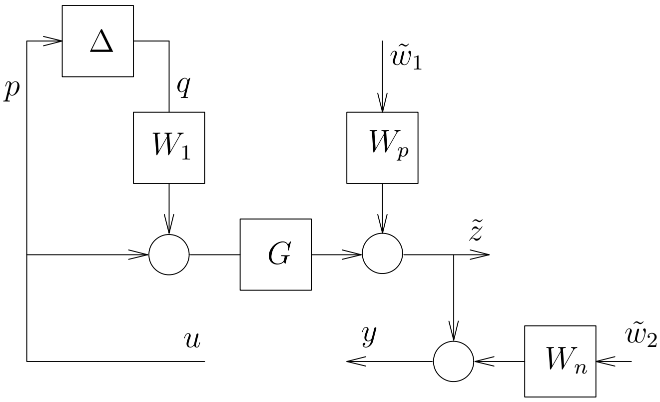

Fig. 2. Spinning satellite interconnection structure

Similarly, the sensor noise frequency content is represented by the transfer function

$$W_n = \frac{12(s + 25)}{5(s + 6000)}.$$

The interconnection structure P has 8 states, 6 outputs and 8 inputs. D_{yu} is a 2-by-2 matrix, that is, the controller uses 2 measurements and has 2 control inputs (figure 2).

The uncertainty structure is $\underline{\Delta} = diag(\delta_1, \delta_2)$ where δ_1 and δ_2 are supposed to be complex numbers with magnitude less than 1.

Here the plant signals $z^T = [q^T, \tilde{z}^T]^T$ and $w^T = [p^T, \tilde{w_1}^T, \tilde{w_2}^T]$ do not have the same dimension. In order to apply the procedure described in Section 2, two extra rows of zeros have been added to C_z, D_{zw} and D_{zu}, so that the augmented plant P, used in the sequel, has a total of 8 inputs and 8 outputs.

4.1 Algorithm initialization

A standard full-order $\mathcal{H}_\infty$ controller, K_0, is first designed with the initial scaling matrix $J = I_2$. The corresponding optimal $\mathcal{H}_\infty$ closed-loop attenuation value is $\gamma = 70.53$. The state-space matrices of the full-order $\mathcal{H}_\infty$ controller K_0 will be used as an initial stating point to our algorithm based on the nonsmooth solver HANSO.

4.2 Nonsmooth optimization vs μ-synthesis

The best $\mathcal{H}_\infty$ attenuation obtained by nonsmooth optimization, when initialized with the full-order $\mathcal{H}_\infty$ controller K_0 and the identity scaling, is $\gamma = 5.11$ (see Table 1) and the sub-optimal scaling matrix, obtained at the end of the optimization process, is

$$J = \begin{pmatrix}0.0513 & 0\\ 0 & 0.0397\end{pmatrix}.$$

In this problem the number of decision variables is 103; 100 decision variables are for the controller, one is for γ and two are for the scaling matrix J. 110 seconds of CPU times (on a 1.9 GHz PC) are necessary for the nonsmooth, nonconvex optimization solver to optimize simultaneously the scaling and controller state-space matrices.

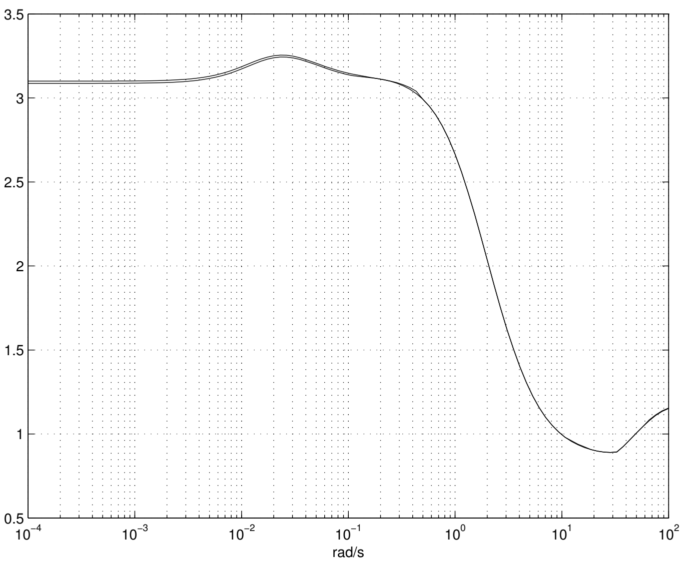

Fig. 3. Upper and lower μ bounds obtained with the nonsmooth controller

The first row in Table 1 is $\| L\mathcal{F}(P,K)L^{-1} \|_\infty$ (i.e. the scaled closed-loop γ attenuation) where L and K are the suboptimal scaling and controller computed by the nonsmooth solver. The second row is the peak value of the structured singular value μ of the closed-loop $\mathcal{F}(P,K)$. More precisely, for a given closed-loop system and an uncertainty structure $\underline{\Delta}$, an upper bound on the peak μ value is

$$\max_k \inf_{D_k \in D_{\underline{\Delta}}} \bar{\sigma}(D_k \mathcal{F}(P,K)(j\omega_k) D_k^{-1}),$$

where $\{\omega_k\}$ is a discrete set of frequency points and where the $D_k s$ are positive definite matrices in the set $D_{\underline{\Delta}}$ whose structure is closely related to the uncertainty structure $\underline{\Delta}$; see Packard and Doyle [1993] for more details. In this case, the peak μ value was computed over the frequency interval $[10^{-4}, 10^2]$ discretized in 100 logarithmically equally spaced points. Lower and upper bounds of the structured singular value corresponding to the closed-loop obtained with the nonsmooth regulator $K(s)$ are shown in figure 3.

It worth noting that closed-loop γ-attenuation and peak μ values are not the same. In general, the peak μ value is less than or equal to the closed-loop γ attenuation value. The reason for this is due to the scaling matrix L which, in our nonsmooth optimization synthesis, is constrained to be constant across the whole imaginary axis, while, in the calculation of the μ upper bounds, the $D_k s$ are optimized at each frequency point (ω_k) and hence are different from each other.

γ	5.11
Peak μ value	3.251

Table 1. μ-synthesis based on nonsmooth optimization

It is instructive to compare the approach of this paper with the standard DK-iteration algorithm of the Matlab Robust Control toolbox (Packard et al. [2004]). The command *dksyn*, which automates the DK-iteration synthesis, is used. To make the comparison sensible, the scaling order is set to 0 in *dksyn*. The frequency range is chosen, as before, as $[10^{-4}, 10^2]$ and contains 100 logarithmically equally spaced points.

iteration	1	2	3	4
γ	70.54	8.56	14.76	17.46
Peak μ value	49.96	4.395	3.256	3.899

Table 2. μ-synthesis based on the DK-iteration procedure

In this experiment, the DK-iteration and our nonsmooth optimization algorithms are initialized with the same initial data, that is the set of parameters corresponding to the full-order $\mathcal{H}_\infty$ controller K_0 and the the identity scaling matrix $J = I_2$.

The DK-iteration results [1] are given in table 2. One can see that DK-iteration exhibits a typical oscillatory behaviour (here after the third iteration; the closed-loop $\mathcal{H}_\infty$ performance and the peak μ values at the fourth iteration are actually bigger than at the third iteration). In this case, the DK-iteration procedure does not provide better controllers and scalings than those directly obtained by nonsmooth optimization. The results obtained by nonsmooth and DK-iteration synthesis are, in terms of μ peak value, very close. This means that a satisfactory μ controller could be computed via DK-iteration, but inevitably this will be done at the expense of using some high order scaling functions (i.e. those interpolating the $D_k s$). However, the optimal γ attenuation value obtained by DK-iteration, with zero order scalings, is 14.76, a value which is almost three times bigger than the γ attenuation achieved directly by nonsmooth optimization ($\gamma = 5.11$). In this example, the algorithm providing the best closed-loop attenuation (with constant scalings) is the controller computed by nonsmooth optimization.

5. CONCLUSIONS

This paper describes a simple μ-synthesis algorithm based on the freely available nonsmooth, nonconvex optimization solver HANSO. It demonstrates the potential of using nonsmooth optimization for μ-synthesis and for other related control synthesis problems. The numerical results given indicate that nonsmooth optimization may provide superior results to those given by the standard DK-iteration procedure. The algorithm of this paper can be further improved. If, for instance, one selects a Jordan canonical form for the controller evolution matrix, then one can reduce significantly the number of decision variables in the optimization program.

Because both DK-iteration and nonsmooth algorithms return local solutions, more numerical experiments are needed to contrast the merits of each approach.

Within this framework, other interesting μ-synthesis problems, such as structured and fixed-order controllers using frequency dependent scalings, could be considered.

[1] Results may vary according to settings in the DK iteration algorithm.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers and Dr. Murray Kerr for their useful comments and suggestions.

REFERENCES

P. Apkarian and R. J Adams. Advanced gain-scheduling techniques for uncertain systems. *IEEE Transactions on Control Systems Technology,*, 6(1):21–32, January 1998.

P. Apkarian and P. Gahinet. A Convex Characterization of Gain-Scheduled $\mathcal{H}_\infty$ Controllers. *IEEE Transactions on Automatic Control,*, 40(5):853–864, May 1995.

P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1):7186, 2006a.

P. Apkarian and D. Noll. Controller Design via Nonsmooth Multidirectional Search. *SIAM J. Control Optim.*, 44 (6):1923 1949, 2007.

P. Apkarian and D. Noll. IQC analysis and synthesis via nonsmooth optimization. *Systems & Control Letters*, 55:971–981, 2006b.

G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith. μ-analysis and synthesis toolbox, version 3. *The MathWorks, Inc.*, January 2001.

D. S. Bernstein. Some Open Problems in Matrix Theory Arising in Linear Systems and Control. *Linear Algebra and its Applications*, (162-164):409–432, 1992.

J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton. HIFOO - A Matlab Package for Fixed-Order Controller Design and H_∞ Optimization. *5th IFAC Symposium on Robust Control Design (ROCOND06), Toulouse, France*, July 5-7 2006.

D. Henrion and M. L. Overton. Maximizing the closed loop asymptotic decay rate for the two-mass-spring control problem. *submitted to Automatica*, 2006.

M. Overton. A Hybrid Algorithm for Non-Smooth Optimization Based on BFGS, Bundle and Gradient Sampling. *http://www.cs.nyu.edu/faculty/overton*, 2006.

A. Packard and J. C. Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, 1993.

A. Packard, G. Balas, M. G. Safonov, R. Chiang, P. Gahinet, and A. Nemirovski. Robust Control Toolbox. *The MathWorks*, Ver. 3.00, 2004.

K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall International, Inc, 1995.