Fast algorithms for solving H_∞-norm minimization problems

Andras Varga

German Aerospace Center
DLR - Oberpfaffenhofen
Institute of Robotics and System Dynamics
D-82234 Wessling, Germany
Andras.Varga@dlr.de

Pablo Parrilo

Control and Dynamical Systems
California Institute of Technology
Mail Stop 107-81
1200 East California Boulevard
Pasadena, CA 91125, USA
pablo@cds.caltech.edu

Abstract

In this paper we propose an efficient computational approach to minimize the H_∞-norm of a transfer-function matrix depending affinely on a set of free parameters. The minimization problem, formulated as a semi-infinite convex programming problem, is solved via a relaxation approach over a finite set of frequency values. In this way, a significant speed up is achieved by avoiding the solution of high order LMIs resulting by equivalently formulating the minimization problem as a high dimensional semidefinite programming problem. Numerical results illustrate the superiority of proposed approach over LMIs based techniques in solving zero order H_∞-norm approximation problems.

1 Introduction

Let $G_\theta(s)$ be a $p \times m$ stable *transfer-function matrix* (TFM) of a linear, continuous time-invariant system with a state space realization $(A, B, C_\theta, D_\theta)$ satisfying

$$G_\theta(s) = C_\theta(sI - A)^{-1}B + D_\theta$$

where C_θ and D_θ are matrices depending affinely on a parameter vector θ. Several approximation problems appearing in model and controller reduction can be formulated as H_∞-norm minimization problems of the form

$$\begin{array}{cc} \min & \|G_\theta(s)\|_\infty \\ \theta & \end{array} \tag{1}$$

For example, in balancing related model reduction problems [10, 9], $G_\theta(s)$ can be defined as the approximation error $G(s) - G_r(s)$ between an n-th order original TFM $G(s) = C(sI - A)^{-1}B + D$ and an r-th order approximation $G_r(s) = C_r(sI - A_r)^{-1}B_r + D_r$, where the entries of matrices C_r and D_r (or B_r and D_r) can be considered as free parameters to be additionally tuned to refine the quality of approximation. Recall that, the H_∞-norm of the approximation error for both methods satisfies [6, 9]

$$\sigma_{r+1} \leq \|G(s) - G_r(s)\|_\infty \leq 2 \sum_{i=r+1}^{n} \sigma_i \tag{2}$$

where $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \geq \sigma_n \geq 0$ are the Hankel-singular values of $G(s)$. Frequently, it is possible to achieve an error bound which nears the lowest achievable error σ_{r+1} by determining the matrices C_r and D_r (or B_r and D_r) which solve the norm minimization problem

$$\begin{array}{cc} \min & \|G(s) - G_r(s)\|_\infty \\ C_r, D_r & \end{array} \tag{3}$$

A more involved approach to compute H_∞-norm approximations involving all four matrices of G_r has been described in [7].

In the Hankel-norm approximation method [6], we are often interested in determining a matrix D_r which also ensures a good H_∞-norm approximation. Because the Hankel-norm is only a seminorm, the approximation $G_r(s)$ is Hankel-norm optimal regardless the chosen feedthrough matrix D_r. It is always possible to choose a D_r such that the upper bound for the approximation error is half of that in (2) [6]. Thus, we can determine an optimal D_r, which minimizes the H_∞-norm of the approximation error, by solving (1) only with D_r as free parameter. In this way, it is often possible to determine an optimal D_r which almost ensures the lowest achievable error σ_{r+1}.

A more difficult L_∞-norm minimization problem appears in the frequency-weighted Hankel-norm approximation [15]. Here, the goal is to minimize a weighted approximation error

$$\|W_1(s)(G(s) - G_r(s))W_2(s)\|_\infty$$

where W_1 and W_2 are given antistable weighting matrices. In the absence of guaranteed upper bounds on the approximation error, it is very desirable to determine the matrices C_r and D_r of the reduced TFM G_r to ensure the lowest achievable frequency-weighted error norm. Frequently good approximations can be achieved by using optimization techniques which ensure error norms very near to the lowest achievable one [15].

All problems discussed above are convex optimization problems and their solutions can be computed by using appropriate techniques. The standard approach to solve such a problem is to convert it first into semi-infinite convex optimization problem which in turn can be equivalently reformulated as a finite dimensional *semidefinite programming* (SDP) problem [3]. Then, this equivalent SDP problem is solved by using, for example, efficient interior point methods [11]. It is typical for model reduction problems, that the order n of the original model $G(s)$ is quite large, often of order of several hundreds and more. Although the computation of the reduced models has an acceptable low computational complexity of order $O(n^3)$, the computation of the optimal D_r (or C_r and D_r) using *linear matrix inequality* (LMI) based solvers implies a high complexity of order $O(n^6)$. Since for large problems the interior point methods grow terribly large and consume huge amount of memory, the solution of dense LMIs on typical workstations is restricted to sizes for n at most 50-100. This makes the computation of an optimal D_r (or C_r and D_r) for larger n either infeasible or undesirable because of too high computational costs.

In this paper we propose an efficient alternative numerical approach to minimize the H_∞-norm of a large McMillan order TFM which depends affinely on a set of free parameters. The solution method is based on solving an equivalent semi-infinite convex optimization problem via a relaxation approach over a finite set of frequency values [12]. A significant speed up is achieved by avoiding the solution of high order LMIs and solving instead, at each iteration, LMIs of order of the number of free parameters. Numerical results obtained by computing zero-order H_∞-norm approximations of linear time-invariant systems indicate the superiority of this approach over the traditional SDP solution.

2 Basic algorithm

We discuss first the solution of the minimization problem (1). Using the definition of the H_∞-norm, this problem is in fact a min-max problem

$$\min_{\theta} \max_{\omega} \|G_\theta(j\omega)\|_2 \tag{4}$$

If we introduce the auxiliary variable

$$\gamma = \max_{\omega} \|G_\theta(j\omega)\|_2$$

then (4) can be rewritten as

$$\begin{array}{c} \min\limits_{\gamma,\theta} \ \gamma \\ \text{subject to } \|G_\theta(j\omega)\|_2 \le \gamma, \ \forall\omega \end{array} \tag{5}$$

This is a semi-infinite convex programming problem which can be solved via an SDP approach by reformulating it as [3]

$$\begin{array}{c} \min\limits_{\gamma,\theta,P} \ \gamma \\ \text{subject to } \left[\begin{array}{ccc} A^TP+PA & PB & C_\theta^T \\ B^TP & -\gamma I_m & D_\theta^T \\ C_\theta & D_\theta & -\gamma I_p \end{array}\right] \le 0 \end{array} \tag{6}$$

where P is symmetric and non-negative definite. This reformulation of the semi-infinite convex optimization problem (5) is very attractive, because it allows to solve a problem with infinite number of constraints as a finite dimensional SDP for which efficient interior-point algorithms exist. Still the computational complexity to solve the LMIs for a n-dimensional model is $O(n^6)$. Thus for many large order problems, as those encountered in model reduction applications, this solution approach is not practical.

A relaxation approach based on an adaptive frequency sampling has been proposed in [12], which allows to reduce in many cases the computational burden to an acceptable level. The basic approach is to solve the semi-infinite convex optimization problem using an *outer approximation* method [13] in combination with exploiting particular features of the algorithms to evaluate the H_∞-norm of linear systems. The relaxation procedure replaces basically the infinite number of constraints in (5) by a finite set of frequencies in a set Ω. This set is updated at each iteration, by adding new frequency points where the constraints are mostly violated. In what follows we give the basic algorithm to solve the semi-infinite programming problem (1) and discuss specific aspects of it.

SIP Algorithm.

1. Set $\omega_0 = 0$, $\Omega = \{\omega_0\}$, and $k = 1$.
2. Solve for θ_k and γ_k the *nonlinear-programming problem* (NLP)

$$\begin{array}{c} \min\limits_{\gamma,\theta} \ \gamma \\ \text{subject to } \|G_\theta(j\omega_i)\|_2 \le \gamma, \ \omega_i \in \Omega. \end{array} \tag{7}$$

3. Compute $\gamma_{k,max} = \|G_{\theta_k}(s)\|_\infty$ and the corresponding ω_k for which $\gamma_{k,max} = \|G_{\theta_k}(j\omega_k)\|_2$.
4. If $\gamma_{k,max} > \gamma_k$, then $\Omega \leftarrow \Omega \cup \{\omega_k\}$, $k \leftarrow k+1$ and go to step 2; else, stop.

At a certain iteration step k, $\Omega = \{\omega_1, \dots, \omega_{k-1}\}$ is the set of frequency points where the infinite set of constraints in (5) is replaced by a finite set of $k-1$ constraints used to solve the standard NLP (7) for the frequency points where the constraints are considered active. The optimization problem to be solved at step 2 is essentially a min-max optimization for which efficient algorithms exist, as for instance, the *sequential-quadratic programming* (SQP) method (see for example [5]). Alternatively, this subproblem can be transformed into a low dimensional SDP problem of the form

$$\begin{array}{ll} \min\limits_{\gamma,\theta} & \gamma \\ \text{subject to} & \begin{bmatrix} \gamma I_p & G_\theta(j\omega_i) \\ G_\theta^*(j\omega_i) & \gamma I_m \end{bmatrix} \geq 0, \ \omega_i \in \Omega \end{array} \tag{8}$$

Here, the dimensions of LMIs are of order of the dimension of the vector θ of free parameters. This dimension is usually much lower than the system order n and therefore these LMIs can be efficiently solved using existing standard tools. Note however, that most standard tools will fail even for moderate values of n when applied directly to problem (5).

The algorithm produces a strictly increasing sequence $\{\gamma_k\}$, representing the worst-case norms at successive iterations. Each new value for γ_k gives automatically a corresponding worst-case frequency ω_k to be added to Ω at the next iteration. This is a byproduct of the algorithm used to evaluate the H_∞-norm of $G_{\theta_k}(s)$ at step 3, as for instance, the quadratically convergent algorithm of [4]. The basic computation in this algorithm is the determination of the eigenvalues of a $2n \times 2n$ Hamiltonian matrix

$$L = \begin{bmatrix} A - BM_{22}^{-1}M_{21} & -BM_{22}^{-1}B^T \\ -M_{11} + M_{12}M_{22}^{-1}M_{21} & -A^T + M_{12}M_{22}^{-1}B^T \end{bmatrix}$$

where

$$M = \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}\right] := \left[\begin{array}{c|c} C_\theta^T C_\theta & C_\theta^T D_\theta \\ \hline D_\theta^T C_\theta & D_\theta^T D_\theta - \gamma^2 I_m \end{array}\right]$$

The norm computation can be speeded up by employing structure exploiting methods to compute Hamiltonian eigenvalues. Algorithms and accompanying software are described in [1]. As worst-case frequency ω_k at step 3, we can safely use the peak frequency value corresponding to the computed H_∞-norm, i.e., the frequency which gives the largest singular value of $G_{\theta_k}(j\omega)$ over all frequency values. Other possible choices are described in [12].

There are several ways for speeding up the basic algorithm. One general approach is to use an adaptive precision technique in solving the subproblems at steps 2 and 3. This means to use larger tolerances at the beginning of iterations and to progressively increase the accuracy of solution as the procedure converges. Another possibility which is appropriate for frequency-weighted problems is to evaluate the H_∞-norm at step 3 using a frequency grid which can be made finer as the iteration progresses in the frequency ranges of interest. Further increase in the efficiency can be obtained by exploiting another feature of the problem. One computation which appears repeatedly, both at step 2 of **SIP Algorithm** as well as in the evaluation of the norm at step 3, is the evaluation of $G_\theta(j\omega)$ for different values of the frequency. By reducing first A to a Hessenberg form, these computations can be done with $O(n^2)$ operations, while for a general A these computations are of order of $O(n^3)$.

3 Discrete-time problems

For a discrete-time setting, we consider the analogous problem to (1)

$$\min_{\theta} \quad \|G_\theta(z)\|_\infty \tag{9}$$

where $G_\theta(z)$ is the Z-transformed (discrete-time) TFM

$$G_\theta(z) = C_\theta(zI - A)^{-1}B + D_\theta.$$

The norm minimization problem (9) can be equivalently formulated as a semi-infinite convex programming problem

$$\begin{array}{ll} \min\limits_{\gamma,\theta} & \gamma \\ \text{subject to} & \|G_\theta(e^{j\omega T})\|_2 \leq \gamma, \ \omega \in [0, \ 2\pi/T] \end{array} \tag{10}$$

where T is the sampling period of the discrete-time system. This problem can be solved via an SDP approach by reformulating it as

$$\begin{array}{ll} \min\limits_{\gamma,\theta,P} & \gamma \\ \text{subject to} & \begin{bmatrix} P - A^TPA & A^TPB & C_\theta^T \\ B^TPA & -\gamma I_m & D_\theta^T \\ C_\theta & D_\theta & -\gamma I_p \end{bmatrix} \leq 0 \end{array} \tag{11}$$

where P is symmetric and non-negative definite. The relaxation algorithm for continuous-time systems can be adapted with obvious modifications to solve discrete-time problems. The new frequency value ω_k generated at step 3 is that value of $\omega \in [0, \ 2\pi/T]$ which corresponds to the peak gain $\gamma_{k,max} = \|G_{\theta_k}(e^{j\omega_k T)}\|_2$ giving the H_∞-norm of $G_{\theta_k}(z)$. Further, the min-max problem to be solved at step 2 can be turned into a low dimensional SDP problem of the form

$$\begin{array}{ll} \min\limits_{\gamma,\theta} & \gamma \\ \text{subject to} & \begin{bmatrix} \gamma I_p & G_\theta(e^{j\omega_i T}) \\ G_\theta^*(e^{j\omega_i T}) & \gamma I_m \end{bmatrix} \geq 0, \ \omega_i \in \Omega \end{array} \tag{12}$$

4 Frequency-weighted problems

We consider the frequency-weighted analog of the problem (1)

$$\min_{\theta} \quad \|W_1(s)G_\theta(s)W_2(s)\|_\infty \tag{13}$$

where, for $i = 1, 2$, $W_i = C_i(sI - A_i)^{-1}B_i + D_i$ are given frequency-weighting TFMs with the corresponding state space representations (A_i, B_i, C_i, D_i) of order n_i. It is not generally possible to convert directly this problem into a nice finite dimensional LMIs based SDP problem. However, the following approach can be always used to turn this problem into an SDP problem. Let θ have r components θ_i, $i = 1, \ldots, r$. Because of the affine dependence of C_θ and D_θ of the elements of the parameter vector θ, it is possible to rewrite

$$\widetilde{G}_\theta(s) := W_1(s)G_\theta(s)W_2(s)$$

in the additive form

$$\widetilde{G}_\theta(s) = G_0(s) + \sum_{j=1}^{r} \theta_j G_j(s)$$

where $G_j(s)$, $j = 1, \ldots, r$ result uniquely. Thus,

$$\widetilde{G}_\theta(s) = \left[\begin{array}{cccc} I_p & \theta_1 I_p & \ldots & \theta_r I_p \end{array}\right] \left[\begin{array}{c} G_0(s) \\ G_1(s) \\ \vdots \\ G_r(s) \end{array}\right]$$

and we can find an equivalent state space representation $(\widetilde{A}, \widetilde{B}, \widetilde{C}_\theta, \widetilde{D}_\theta)$ satisfying

$$\widetilde{G}(s) = \widetilde{C}_\theta(sI - \widetilde{A})^{-1}\widetilde{B} + \widetilde{D}_\theta$$

Note that a state-space representation of order $n + n_1 + n_2$ can be always constructed for $\widetilde{G}(s)$. To solve the convex optimization problem

$$\min_{\theta} \quad \|\widetilde{G}_\theta(s)\|_\infty \tag{14}$$

the same approaches can be employed as those presented in section 2 by working with $\widetilde{G}_\theta(s)$ instead of $G_\theta(s)$. A similar approach can be used also in the discrete-time case.

5 Numerical examples

Example 1: This is example 3 of [15] of a SISO system with the transfer function

$$G(s) = \frac{0.05(s^7+801s^6+1024s^5+599s^4+451s^3+119s^2+49s+5.55)}{s^7+12.6s^6+53.48s^5+90.94s^4+71.83s^3+27.22s^2+4.75s+0.3}$$

We want to approximate $G(s)$ by $G_r(s)$ such that the relative error

$$\|G^{-1}(s)\left(G(s) - G_r(s)\right)\|_\infty \tag{15}$$

is as small as possible. To solve this frequency-weighted model reduction problem, we used the frequency-weighted Hankel-norm approximation method of Latham and Anderson (LA) [8]. Note that for this method, the frequency-weighted approximation error for an r-th order approximation, is bounded from below by σ_{r+1}, where σ_i denotes the i-th largest Hankel singular value of the stable projection of $G^{-1}(-s)G(s)$.

The computed singular values $\{\sigma_1, \ldots, \sigma_7\}$ are

$$\{0.9997,\;\; 0.9982,\;\; 0.9734,\;\; 0.7146,\;\; 0.5584,\;\; 0.0026,\;\; 0\}$$

The third order approximation computed with the LA-method is

$$G_r(s) = \frac{0.08573s^3 + 10.48s^2 + 0.08765s + 2.367}{s^3 + 1.833s^2 + 34.62s + 8.843}$$

and the corresponding approximation error (15) is 0.8417. The resulting state-space realization of $G_r(s)$ is

$$A_r = \begin{bmatrix} -0.7871 & 5.0560 & -1.6967 \\ -6.6439 & -0.7871 & -1.5753 \\ 0 & 0 & -0.2585 \end{bmatrix}, \; B_r = \begin{bmatrix} -6.4222 \\ -6.0968 \\ -0.3095 \end{bmatrix},$$

$$C_r = \begin{bmatrix} -0.7766 & -0.8628 & -0.2275 \end{bmatrix}, \quad D_r = 0.0857$$

By using D_r as free parameter, we solved the corresponding zero-th order frequency-weighted H_∞-norm approximation problem, to determine the optimum $\widetilde{D}_r$ which minimizes the frequency-weighted approximation error. Practical convergence of the **SIP Algorithm** has been obtained in 6 iterations and with the new $\widetilde{D}_r = 0.0868$, the error norm has been reduced to 0.811. By including also the elements of C_r among the optimization variables, we obtained convergence in 8 iterations to the new optimal matrices

$$\widehat{C}_r = \left[\begin{array}{ccc} -0.7710 & -0.8569 & -0.2279 \end{array}\right], \quad \widehat{D}_r = 0.0844$$

for which the corresponding error norm was 0.7199. This value is very close to the best achievable value $\sigma_4 = 0.7146$. In Figure 1 we present the relative error achieved with different approximations. It is easy to observe the almost uniformly flat relative error achieved by fitting both C_r and D_r.

Example 2: We generated various random system examples to test the convergence properties of the proposed method. We solved problems of form (1) by minimizing the H_∞-norm of $G(s) - D_r$, where the matrices of a realization of $G(s)$ have been randomly generated. The optimal computed D_r represents a zero-th order H_∞-norm approximation of $G(s)$.

All computations have been done on a Pentium III 700 MHz machine running Windows 98. To experiment with the **SIP Algorithm**, we employed MATLAB 5.3. For the solution of the LMIs (8) at step 2

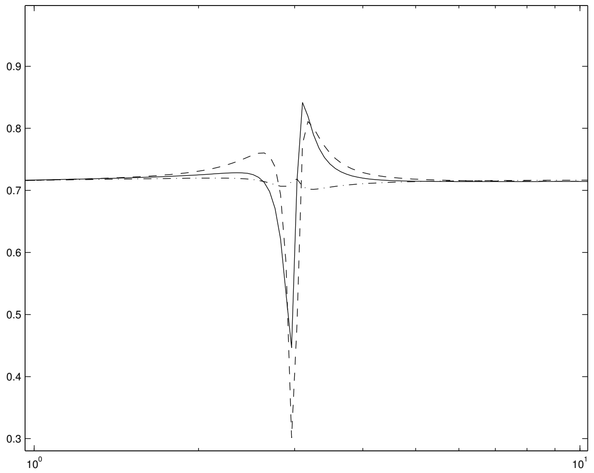

Figure 1: Relative error $|G^{-1}(G - G_r)|$: LA method (solid), optimal $\widetilde{D}_r$ (dashed), optimal $[\widehat{C}_r, \widehat{D}_r]$ (dashdot).

we used the MATLAB toolbox SeDuMi [14] for solving optimization problems over symmetric cones. This package allows to solve SDP problems with linear constraints, quasiconvex-quadratic and positive semidefiniteness constraints. Complex valued entries are allowed and large scale optimization problems are efficiently solved by exploiting sparsity. For the computation of the H_∞-norm at step 3 we used the standard H_∞-norm *m*-function `norm` provided in the MATLAB Control Toolbox version 4.2.

In the Tables 1 and 2 we have timing and convergence results for randomly generated SISO and MIMO stable systems, respectively, with orders ranging from $n = 2$ to $n = 128$. We have used MIMO systems with 3 inputs and 3 outputs. The iterations have been initialized by setting $\Omega = \{0, \omega_{peak}, \infty\}$, where ω_{peak} is the peak-frequency corresponding to the H_∞-norm of $G(s)$. The timing results are only intended to assess the relative computational burden as the dimension of the problem varies. Besides the resulting average times given in variable $\overline{T}$, we also show the average number of iterations $\overline{N}_{conv}$ performed until convergence with an accuracy tolerance of 5 decimal digits, as well as the maximum N_{max} and minimum N_{min} number of iterations for each set of examples. The average times and iterations numbers have been determined over 10 random test examples for each dimension.

n	2	4	8	16	32	64	128
$\overline{T}$ (sec)	0.62	0.53	0.63	1.02	1.2	5.9	77.02
$\overline{N}_{conv}$	4.1	3.6	4.0	5.2	3.9	4.6	4.6
N_{max}	8	8	7	8	7	8	9
N_{min}	1	1	1	2	3	1	2

Table 1: Results with **SIP Algorithm** for SISO systems.

n	2	4	8	16	32	64	128
$\overline{T}$ (sec)	0.5	1.25	2.59	2.83	5.3	12.6	165.6
$\overline{N}_{conv}$	2.4	4.9	7.5	7.6	9.0	8.4	9.1
N_{max}	5	8	12	11	14	14	15
N_{min}	1	2	1	1	4	5	4

Table 2: Results with **SIP Algorithm** for MIMO systems.

The results in these tables indicate that problems of moderate size (with n up to a few hundreds) can be solved by the proposed method with a still acceptable computational effort. It is interesting to note that, for the larger values of n, the times in Tables 1 and 2 reflect almost entirely the computational costs of evaluating the H_∞-norm at step 3 of the **SIP Algorithm**. For example, for $n = 128$, the times for computing the solutions of LMIs (8) was about 1% of the total time, while 98% of time was spent computing norms.

We were able to reduce the times in the tables by factors up to 2–4, by using instead the MATLAB *m*-function `norm` to evaluate H_∞-norms, a fast *mex*-function `linorm` developed recently within the NICONET project [2]. The speedup has been mainly achieved by using the symplectic method of [1] to compute the eigenvalues of Hamiltonian matrices instead the standard QR-iteration based method to compute eigenvalues. Occasionally, significant improvements also resulted by employing the peak frequency computed at step 3 of the **SIP Algorithm** in one iteration to initialize the norm computation in the next iteration. This nice feature is provided by the newly developed *mex*-function `linorm`.

6 Conclusions

We have shown that several affine approximation problems encountered in model reduction can be solved using a fast algorithm which avoids the conversion to an equivalent high order SDP formulation. Since the solution of SDP problems via LMIs is not feasible even for moderately large dimensions, the new method offers a viable alternative to solve this class of approximation problems. Important speedup can be achieved by fully exploiting the structure of the underlying optimization problem. Preliminary experimental results indicate a good potential of the proposed approach to address other similar problems in the control theory. An open aspect related to the **SIP algorithm** is the development of a rigorous theory for its convergence rate, independently of or in conjunction with adaptive precision computations to solve subproblems.

References

[1] P. Benner, R. Byers, and E. Barth. Algorithm 800: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices I: The square-reduced method. *ACM Transactions on Mathematical Software*, 26:49–77, 2000.

[2] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT – a subroutine library in systems and control theory. In B. N. Datta, editor, *Applied and Computational Control, Signals and Circuits*, vol. 1, pp. 499–539. Birkhäuser, 1999.

[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, *SIAM Studies in Applied Mathematics*, vol. 15, SIAM, Philadelphia, PA 19101, USA, 1994.

[4] N.A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H_∞-norm of a transfer function. *Systems & Control Letters*, 14:287–293, 1990.

[5] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimisation.* Academic Press, London, 1981.

[6] K. D. Glover. All optimal Hankel-norm approximation of linear multivariable systems and their L^∞-error bounds. *Int. J. Control*, 39:1115–1193, 1984.

[7] A. Helmerson. Model reduction using LMIs. In *Proc. of 33rd CDC, Lake Buena Vista, FL*, pp. 3217–3222, 1994.

[8] G. A. Latham and B. D. O. Anderson. Frequency-weighted optimal Hankel norm approximation of stable transfer functions. *Systems* & *Control Lett.*, 5:229–236, 1985.

[9] Y. Liu and B. D. O. Anderson. Singular perturbation approximation of balanced systems. *Int. J. Control*, 50:1379–1405, 1989.

[10] B. C. Moore. Principal component analysis in linear system: controllability, observability and model reduction. *IEEE Trans. Autom. Control*, 26:17–32, 1981.

[11] Yu. E. Nesterov and A. S. Nemirovsky. *Interior–Point Polynomial Algorithms in Convex Programming : Theory and Algorithms*, *Studies in Applied Mathematics*, vol. 13, SIAM Publications, Philadelphia, PA 19101, USA, 1994.

[12] P. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, Californian Institute of Technology, 2000.

[13] E. Polak. *Optimization : Algorithms and Consistent Approximations*, *Applied Mathematical Sciences*, vol. 124, Springer, New York, 1997.

[14] J. F. Sturm. Using SeDuMi 1.02: A MATLAB Toolbox for optimization over symmetric cones. *Optimization Meth. & Soft.*, vol. 11&12, pp. 625–653, 1999.

[15] K. Zhou. Frequency-weighted $\mathcal{L}_\infty$ norm and optimal Hankel norm model reduction. *IEEE Trans. Autom. Control*, 40:1687–1699, 1995.