Efficient Solution of Linear Matrix Inequalities for Integral Quadratic Constraints

Anders Hansson
Department of Signals, Sensors and Systems
Royal Institute of Technology
SE–100 44 Stockholm, Sweden
`anders.hansson@s3.kth.se`

Lieven Vandenberghe
Electrical Engineering Department
University of California, Los Angeles
Los Angeles, CA 90095-1594, USA
`vandenbe@ee.ucla.edu`

Abstract

In this article is discussed how to implement an efficient interior-point algorithm for the semi-definite programs that result from integral quadratic constraints. The algorithm is a primal-dual potential reduction method, and the computational effort is dominated by a least-squares system that has to be solved in each iteration. The key to an efficient implementation is to utilize iterative methods and the specific structure of integral quadratic constraints. The algorithm has been implemented in Matlab. To give a rough idea of the efficiencies obtained, it is possible to solve problems resulting in a linear matrix inequality of dimension 130 × 130 with approximately 5000 variables in about 10 minutes on a lap-top. Problems with approximately 20000 variable and a linear matrix inequality of dimension 230 × 230 are solved in a few hours. It is not assumed that the system matrix has no eigenvalues on the imaginary axis, nor is it assumed that it is Hurwitz.

Key Words: Linear Matrix Inequality, Integral Quadratic Constraint, Semi-Definite Program, Interior-Point Method, Robust Control

1 Introduction

A unified approach to robustness analysis in control with respect to nonlinearities, time variations, and uncertain parameters was presented by Megretski and Rantzer in [6] using so called Integral Quadratic Constraints (IQCs). Applying the Kalman-Yakubovich-Popov (KYP) lemma, e.g. [9], they show how IQCs can be reformulated as Linear Matrix Inequalities (LMIs), e.g. [1]. The corresponding optimization problem is a Semi-Definite Program (SDP). Efficient interior-point algorithms for general SDPs have been presented in [7], but these algorithms are still not efficient enough for the very large SDPs that result from IQCs. It is possible to overcome this by utilizing the special structure inherited from the IQCs. It is the objective to show how this can be done using ideas similar to the ones presented in [2, 11]. Previous work on implementing efficient solvers for these SDPs are based on cutting-plane algorithms, [5] and outer approximation methods, [8]. A more detailed description of our work is presented in [3].

2 Primal and Dual Problems

The following optimization problem in the symmetric $n_s \times n_s$ matrix P and the vector $\mu \in \mathbf{R}^p$ can be related to IQCs:

$$\begin{array}{ll} \text{minimize} & -x_0^T P x_0 + \sum_{i=1}^{p} \mu_i \gamma_i \\ \text{subject to} & \bar{A} P \bar{B} + \bar{B}^T P \bar{A}^T + M \geq 0 \\ & \mu \geq 0 \end{array}$$

where $\bar{A}^T = \begin{bmatrix} I & 0 \end{bmatrix}$, $\bar{B} = \begin{bmatrix} A & B \end{bmatrix}$, and where $M = M_0 + \sum_{i=1}^{p} \mu_i M_i$ is a symmetric matrix. It will be assumed that (A,B) is stabilizable. The stabilizability will make it easy to find strictly feasible initial points for the algorithm. The corresponding dual problem reads

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr} M_0 Z_1 \\ \text{subject to} & \bar{B} Z_1 \bar{A} + \bar{A}^T Z_1 \bar{B}^T + x_0 x_0^T = 0 \\ & \mathbf{Tr} M_i Z_1 + z_{2i} - \gamma_i = 0, \quad i = 1, 2, \ldots, p \\ & Z = Z_1 \oplus_{i=1}^{p} z_{2i} \geq 0 \end{array}$$

Most primal-dual algorithms for solving the primal and dual SDPs are what is called feasible, i.e. every iterate (P^k, μ^k, Z^k) is strictly feasible. Specifically (P^0, μ^0, Z^0) has to be strictly feasible. It is not in any way trivial to find such an initial point, and therefore we will introduce some artificial bounds. This is commonly known as the 'big-M' method, [12]. This will modify the primal and dual problems, and thereby make it easier to find strictly feasible initial points. More specifically the following bounds are introduced on P, μ, and Z_1: $\mathbf{Tr} P \geq -\tilde{M}$; $\sum_{i=1}^{p} \mu_i \leq \tilde{M}$; $\mathbf{Tr} Z_1 \leq \bar{M}$, where $\bar{M}, \tilde{M} > 0$. The primal-dual potential reduction method used in this article is due to Nesterov and Todd, e.g. [10], and it is the same method that is used in SP, [13]. The main computational burden lies in computing the search direction, which turns out to be equivalent to solving a least-squares problem. The corresponding normal equations can be solved using many different methods. Iterative methods

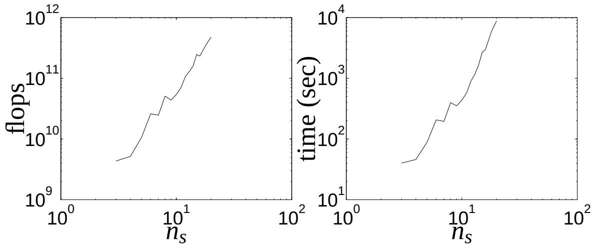

Figure 1: Flop count and CPU-time.

are very favorable, since it turns out that if a good preconditioner is used very few iterations have to be made. We use the CG method for simplicity. The preconditioner is a block Jacobi method. Moreover the CG method is stopped before it has converged. This will result in inexact search directions, which by solving a Lyapunov equation can be made feasible. Fore details see [3].

3 Numerical Example

The computations have been done on an IBM ThinkPad 570E running Matlab under Linux. We will consider the same example as Example 2 in [8] except for the numerical values. The value of n_s is 50. The solution can be computed with the proposed algorithm using 12.4 Gflops in 102.5 seconds. In [8] the example is solved in less than 120 seconds. It is of course difficult to say what algorithms is the fastest without doing a comparison on the same computer and with exactly the same example. To investigate how the computational complexity depends on n_s a series of problems similar to the above one were solved. In Figure 1 the flop count is plotted versus n_s. It is seen that the flop-count is of order less than n_s^3. The same is true for the computational time. It is seen that a problem with $n_s = 100$ is solved in less than 10 minutes. Notice that for $n_s = 200$ we solve an SDP where the number of entries in P is 40000 and the dimension of the LMI is about 230×230. This means that the number of primal variables are about 20000. The computational time for this example is less than two and a half hour.

4 Conclusions

In this article some preliminary results have been presented on how to efficiently solve SDPs for IQCs. The algorithm used is the one proposed by Nesterov and Todd. The key to obtain high speed is the use of approximate search directions. In an example it has been seen that interior-point methods perform just as well as recently proposed cutting-plane methods. It should be stressed that for those methods only μ is computed. The matrix P has to be computed separately by solving an algebraic Riccati equation once μ has been found. This Riccati equation is potentially indefinite, i.e. M might be indefinite. Standard software for algebraic Riccati equations, such as what is available in Matlab, is not applicable, and more advanced solvers, such as the one proposed in [4], has to be used. The flop-count and computational time is of order less than n_s^3. This is extremely good since standard algorithms for solving algebraic Riccati equations, such as the Schur form method, are of order n_s^3. Moreover we are able to solve problems with as many as 20000 primal variables and LMI constraints of dimension above 200×200.

References

[1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, 1994.

[2] S Boyd, L. Vandenberghe, and M. Grant. Efficient convex optimization for engineering design. In *Proceedings of the IFAC Symposium on Robust Control Design*, pages 14–23, Rio de Janeiro, Brazil, 1994.

[3] A. Hansson and L. Vandenberghe. Efficient solution of linear matrix inequalities for integral quadratic constraints. Internal Report IR–S3–REG–0002, Division of Automatic Control, Royal Institute of Technology, August 2000.

[4] V. Ionescu, C. Oara, and M. Weiss. *Generalized Riccati Theory and Robust Control*. John Wiley & Sons, Chichester, 1999.

[5] C.-Y. Kao, A. Megretski, and U. Jönsson. An algorithm for solving optimization problems involving special frequency dependent lmis. In *Proceedings of the 20th American Control Conference*, Chicago, IL, 2000.

[6] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, AC-42(6):819–830, 1997.

[7] Y. Nesterov and A. Nemirovsky. *Interior-Point Polynomial Methods in Convex Programming*. SIAM, Philadelphia, 1994.

[8] P. A. Parrilo. On the numerical solution of LMIs derived from the KYP lemma. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona, 1999.

[9] A. Rantzer. On the Kalman-Yakubovish-Popov lemma. *Systems & Control Letters*, 28:7–10, 1996.

[10] M. J. Todd, K. C Toh, and R. H Tütüncü. On the Nesterov-Todd direction in semidefinite programming. *SIAM Journal on Optimization*, 8(3):769–796, 1998.

[11] L. Vandenberghe and S. Boyd. A primal-dual potential reduction method for problems involving matrix inequalities. *Mathematical Programming*, 69:205–236, 1995.

[12] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[13] L. Vandenberghe and S. Boyd. *SP—Software for Semidefinite Programming—User's Guide*. University of California Los Angeles and Stanford University, 1998.