Technische Universität Berlin
Institut für Mathematik

Evaluation of Numerical Methods for Discrete-Time $\mathcal{H}_\infty$ Optimization

Volker Mehrmann and Petko Petkov

Preprint 08–2005

Preprint-Reihe des Instituts für Mathematik
Technische Universität Berlin

Report 08–2005 March 2005

Evaluation of Numerical Methods for Discrete-Time $\mathcal{H}_\infty$ Optimization

V. Mehrmann[*] and **P. Petkov**[†]

04.02.2005

Abstract

We compare the numerical properties of the different numerical methods for solving the $\mathcal{H}_\infty$ optimization problems for linear discrete-time systems. It is shown that the methods based on the solution of the associated discrete-time algebraic Riccati equation may be unstable due to an unnecessary increase in the condition number and that they have restricted application for ill-conditioned and singular problems. The experiments confirm that the numerical solution methods that are based on the solution of a Linear Matrix Inequality (LMI) are a much more reliable although much more expensive numerical technique for solving $\mathcal{H}_\infty$ optimization problems. Directions for developing high-performance software for $\mathcal{H}_\infty$ optimization are discussed.

Key Words: $\mathcal{H}_\infty$-optimization, $\mathcal{H}_\infty$-control, discrete-time system, linear matrix inequality, discrete-time algebraic Riccati equation

[*]Institut für Mathematik, MA 4-5, Technische Universität Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany `mehrmann@math.tu-berlin.de`. This author was partially supported by *Deutsche Forschungsgemeinschaft*, Research Grants Me 790/11-3 and the DFG Research Center MATHEON, Mathematics for Key Technologies, in Berlin.

[†]Department of Automatics, Technical University of Sofia, 1756 Sofia, Bulgaria Supported by Deutscher Akademischer Austauschdienst

1 Introduction

The discrete-time $\mathcal{H}_\infty$ optimization problem is usually not presented in detail in the control theory literature [10, 17] due to the fact that it is mathematically equivalent to an appropriate continuous-time $\mathcal{H}_\infty$-optimization problem through a simple bilinear transformation. However, from the point of view of numerical methods, the implementation of bilinear transformations (especially for high order systems) is accompanied with severe difficulties, due to a possible increase in sensitivity as well as high computational costs. For this reason it is more appropriate to use numerical methods that are able to directly solve the discrete-time optimal $\mathcal{H}_\infty$ optimization problem. Several such methods are available in the literature, but their numerical properties have not been studied and compared in detail. For this reason, in practice one faces the difficult task to choose the best software yielding the best solution for a given discrete-time $\mathcal{H}_\infty$ optimization problem.

In this report we compare the numerical properties of the different methods for solving the $\mathcal{H}_\infty$-optimization problems for linear discrete-time systems. We show that the methods based on the solution of algebraic Riccati equations introduce several numerical difficulties. In particular, we show that this approach is not always stable, due to the fact that it introduces unnecessary sensitivity and may lead to ill-conditioned or even singular subproblems.

Our extensive experiments show that the solution methods that are based on the LMI (linear matrix inequality) formulation is a much more reliable, although more expensive, numerical technique for solving $\mathcal{H}_\infty$ optimization problems.

This leads to the conclusion that a numerically reliable and efficient method for the discrete-time $\mathcal{H}_\infty$-optimization is still lacking. Directions for developing high-performance software for such optimization problems are briefly discussed.

The report is organized as follows. In Section 2 we give the standard definition of the discrete-time $\mathcal{H}_\infty$ optimization problem. The available numerical methods for solving this problem are surveyed in Section 3. They are divided into three groups involving implementation of bilinear transformation, solution of Riccati equations and solutions of LMIs. A Riccati equations-based method is described in some detail in Section 4, while in Section 5 we present an LMI-based method. The available software for discrete-time $\mathcal{H}_\infty$ optimization is listed in Section 6. This software includes functions from MATLAB[1] toolboxes as well as mex-files based on SLICOT Fortran 77 routines [16]. Results from the numerical evaluation of the different methods for a restricted class of examples are presented in Section 7. Finally, in Section 8 we give some conclusions derived from the numerical experiments and we discuss directions for further development of high-performance software for discrete-time $\mathcal{H}_\infty$ optimization.

2 The discrete-time $\mathcal{H}_\infty$-optimization problem

Consider a linear discrete-time system, described by the equations

$$\begin{aligned} x_{k+1} &= Ax_k + B_1 w_k + B_2 u_k, \\ z_k &= C_1 x_k + D_{11} w_k + D_{12} u_k, \\ y_k &= C_2 x_k + D_{21} w_k + D_{22} u_k, \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state vector, $w_k \in \mathbb{R}^{m_1}$ is a exogenous input vector (the disturbance), $u_k \in \mathbb{R}^{m_2}$ is the control input vector, $z_k \in \mathbb{R}^{p_1}$ is a controlled vector, and $y_k \in \mathbb{R}^{p_2}$ is a measurement vector. The transfer function matrix of the system is denoted by

$$\begin{aligned} P(z) &= \begin{bmatrix} P_{11}(z) & P_{12}(z) \\ P_{21}(z) & P_{22}(z) \end{bmatrix} \\ &\hat{=} \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \\ &\hat{=}: \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]. \end{aligned} \tag{2}$$

The $\mathcal{H}_\infty$*-suboptimal discrete-time control problem* is to find an internally stabilizing controller $K(z)$ such that, for a pre-specified positive value of γ, the inequality

$$\|F_\ell(P,K)\|_\infty < \gamma \tag{3}$$

is satisfied, where $F_\ell(P,K)$ is the lower linear fractional transformation (LFT) on $K(z)$, equal to the closed-loop transfer function $T_{zw}(z)$ from w to z,

$$T_{zw}(z) := F_\ell(P,K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$

[1]MATLAB is a trade mark of MathWorks, Inc.

In the $\mathcal{H}_\infty$*-optimization control problem* one tries to find the infimum of γ (further denoted by γ_{opt}) which satisfies (3). This infimum is difficult to find analytically in the general case. It is usually computed numerically by a root-finding procedure involving some method for suboptimal design. The accuracy of achieving the infimum is controlled by an appropriately chosen tolerance. The solution of the $\mathcal{H}_\infty$-optimization control problem corresponds to the best disturbance attenuation at the controlled output of the closed-loop system.

3 Numerical methods for $\mathcal{H}_\infty$-suboptimal design of discrete-time controllers

There are three groups of methods for solving discrete-time $\mathcal{H}_\infty$-optimization problems.

- *Methods based on a bilinear transformation to a continuous-time problem [1, 4].*

  These methods allow to solve the discrete-time $\mathcal{H}_\infty$-optimization problem via the algorithms and software for solving continuous-time problems. They implement a bilinear transformation and its inverse so that the accuracy of the final solution depends on the condition number of this transformation. Typically such methods perform badly for high order problems and should be considered unreliable from the numerical point of view. Note that it is possible to derive formulas for the discrete-time controller implementing analytically the bilinear transformation [12]. Unfortunately, these formulas are very complicated and we were unable to find a stabilizing controller when applying these formulas to our examples.

- *Riccati equations based methods [10, 9].*

  These methods include the solution of two matrix algebraic Riccati equations and represent an efficient way for solving the $\mathcal{H}_\infty$-suboptimal problem requiring a volume of computational operations proportional to n^3. Their implementation, however, is restricted only to *regular plants*, i.e. plants for which the matrices D_{12} and D_{21} are of full rank and the transfer functions from controls to controlled outputs ($P_{12}(z)$) and from disturbance to measured outputs ($P_{21}(z)$) have no invariant zeros on the unit circle. Although various extensions are proposed for the *singular plants*, the numerical difficulties in such cases are not overcame. Also, for γ approaching the optimum value γ_{opt} the Riccati equations become ill-conditioned which may drastically affect the accuracy of the computed controller matrices.

- *LMI-based methods [7, 6].*

  These methods are based on the solution of three or more linear matrix ineqalities (LMIs) derived from Lyapunov based criteria. The main advantage of these methods is that they can be applied without difficulties to singular plants, the only assumption being the stabilizability and detectability of the triple (A, B_2, C_2). The LMI approach yields a finite-dimensional parametrization of all $\mathcal{H}_\infty$-controllers which allows to exploit the remaining freedom for controller order reduction and for handling additional constraints on the closed-loop performance. The LMIs are solved by convex optimization algorithms which require a volume of computational operations proportional to n^6. This fact restricts the implementation of LMI-based methods to relatively low order plants in contrast to the Riccati equations based methods.

In the next two sections we consider in more detail two methods based on Riccati equations solution and LMIs solution, respectively, which in our view represent the state-of-the-art in solving the discrete-time $\mathcal{H}_\infty$-suboptimal problem.

4 Riccati equation based methods

In this section we briefly present the method for the design of suboptimal $\mathcal{H}_\infty$-controllers that is based on the solution of discrete-time algebraic Riccati equations, as proposed in [10]. This method (which we call *Riccati-method*) is derived under the following assumptions.

A1 (A, B_2) is stabilizable and (C_2, A) is detectable;

A2 $\begin{bmatrix} A - e^{j\Theta} I_n & B_2 \\ C_1 & D_{12} \end{bmatrix}$ has full column rank for all $\Theta \in [0, 2\pi)$;

A3 $\begin{bmatrix} A - e^{j\Theta} I_n & B_1 \\ C_2 & D_{21} \end{bmatrix}$ has full row rank for all $\Theta \in [0, 2\pi)$.

We shall assume also that a loop-shifting transformation that enables to set $D_{22} = 0$ has been carried out. We shall return to the general case $(D_{22} \neq 0)$ later on.

Note that the method under consideration does not involve reduction of the matrices D_{12} and D_{21} to some special form, as it is usually required in the design of continuous-time $hinf$-controllers [5].

Let

$$\bar{C} = \begin{bmatrix} C_1 \\ 0 \end{bmatrix}, \quad \bar{D} = \begin{bmatrix} D_{11} & D_{12} \\ I_{m_1} & 0 \end{bmatrix}$$

and define the matrices

$$J = \begin{bmatrix} I_{p_1} & 0 \\ 0 & -\gamma^2 I_{m_1} \end{bmatrix}, \quad \hat{J} = \begin{bmatrix} I_{m_1} & 0 \\ 0 & -\gamma^2 I_{m_2} \end{bmatrix}, \quad \tilde{J} = \begin{bmatrix} I_{m_1} & 0 \\ 0 & -\gamma^2 I_{p_1} \end{bmatrix}.$$

Let X_∞ be the solution to the discrete-time algebraic Riccati equation

$$X_\infty = \bar{C}^T J \bar{C} + A^T X_\infty A - L^T R^{-1} L, \tag{4}$$

where

$$R = \bar{D}^T J \bar{D} + B^T X_\infty B =: \begin{bmatrix} R_1 & R_2^T \\ R_2 & R_3 \end{bmatrix},$$

$$L = \bar{D}^T J \bar{C} + B^T X_\infty A =: \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}.$$

Assume that there exists an $m_2 \times m_2$ matrix V_{12} such that

$$V_{12}^T V_{12} = R_3$$

and an $m_1 \times m_1$ matrix V_{21} such that

$$V_{21}^T V_{21} = -\gamma^{-2} \nabla, \quad \nabla = R_1 - R_2^T R_3^{-1} R_2 < 0.$$

Define the matrices

$$\begin{bmatrix} A_t & \tilde{B}_t \\ C_t & \tilde{D}_t \end{bmatrix} =: \left[\begin{array}{c|cc} A_t & \tilde{B}_{t_1} & \tilde{B}_{t_2} \\ \hline C_{t_1} & \tilde{D}_{t_{11}} & \tilde{D}_{t_{12}} \\ C_{t_2} & \tilde{D}_{t_{21}} & \tilde{D}_{t_{22}} \end{array}\right] =$$

$$\left[\begin{array}{c|cc} A - B_1 \nabla^{-1} L_\nabla & B_1 V_{21}^{-1} & 0 \\ \hline V_{12} R_3^{-1} (L_2 - R_2 \nabla^{-1} L_\nabla) & V_{12} R_3^{-1} R_2 V_{21}^{-1} & I \\ C_2 - D_{21} \nabla^{-1} L_\nabla & D_{21} V_{21}^{-1} & 0 \end{array}\right],$$

where

$$L_\nabla = L_1 - R_2^T R_3^{-1} L_2.$$

Let Z_∞ be the solution to the discrete-time algebraic Riccati equation

$$Z_\infty = \tilde{B}_t \hat{J} \tilde{B}_t^T + A_t Z_\infty A_t^T - M_t S_t^{-1} M_t^T, \tag{5}$$

in which

$$
\begin{aligned}
S_t &= \tilde{D}_t \hat{J} \tilde{D}_t^T + C_t Z_\infty C_t^T =: \begin{bmatrix} S_{t_1} & S_{t_2} \\ S_{t_2}^T & S_{t_3} \end{bmatrix}, \\
M_t &= \tilde{B}_t \hat{J} \tilde{D}_t^T + A_t Z_\infty C_t^T =: [M_{t_1} M_{t_2}].
\end{aligned}
$$

In the following, we refer to equations (4) and (5) as *X-Riccati equation* and *Z-Riccati equation*, respectively.

As shown in [10], a stabilizing controller that satisfies

$$
\|F_\ell(P,K)\|_\infty < \gamma
$$

exists, if and only if

1. There exists a solution to the Riccati equation (4) satisfying

$$
X_\infty \geq 0, \;\; \nabla < 0,
$$

such that $A - BR^{-1}L$ is asymptotically stable.

2. There exists a solution to the Riccati equation (5) such that

$$
Z_\infty \geq 0, \;\; S_{t_1} - S_{t_2} S_{t_3}^{-1} S_{t_2}^T < 0,
$$

with $A_t - M_t S_t^{-1} C_t$ asymptotically stable.

In this case, a controller that achieves the objective is

$$
\begin{aligned}
\hat{x}_{k+1} &= A_t \hat{x}_k + B_2 u_k + M_{t_2} S_{t_3}^{-1} (y_k - C_{t_2} \hat{x}_k), \\
V_{12} u_k &= -C_{t_1} \hat{x}_k - S_{t_2} S_{t_3}^{-1} (y_k - C_{t_2} \hat{x}_k),
\end{aligned}
$$

which yields

$$
K_0 = \left[\begin{array}{c|c} A_t - B_2 V_{12}^{-1} (C_{t_1} - S_{t_2} S_{t_3}^{-1} C_{t_2}) - M_{t_2} S_{t_3}^{-1} C_{t_2} & -B_2 V_{12}^{-1} S_{t_2} S_{t_3}^{-1} + M_{t_2} S_{t_3}^{-1} \\ \hline -V_{12}^{-1} (C_{t_1} - S_{t_2} S_{t_3}^{-1} C_{t_2}) & -V_{12}^{-1} S_{t_2} S_{t_3}^{-1} \end{array} \right].
$$

This is the so called *central controller* which is widely used in practice. In the following we only consider the computation of the central suboptimal controller.

Consider now the general case when $D_{22} \neq 0$. Suppose that

$$
\hat{K} = \left[\begin{array}{c|c} \hat{A}_k & \hat{B}_k \\ \hline \hat{C}_k & \hat{D}_k \end{array} \right]
$$

is a stabilizing controller for D_{22} set to zero, and satisfies

$$\|F_\ell \left(P - \left[\begin{array}{cc} 0 & 0 \\ 0 & D_{22} \end{array} \right], \hat{K} \right) \|_\infty < \gamma.$$

Then [15]

$$\begin{array}{rcl} F_\ell(P, \hat{K}(I + D_{22}\hat{K})^{-1}) & = & P_{11} + P_{12}\hat{K}(I + D_{22}\hat{K} - P_{22}\hat{K})^{-1}P_{21} \\ & = & F_\ell \left(P - \left[\begin{array}{cc} 0 & 0 \\ 0 & D_{22} \end{array} \right], \hat{K} \right). \end{array}$$

In this way a controller $\hat{K}$ for

$$P - \left[\begin{array}{cc} 0 & 0 \\ 0 & D_{22} \end{array} \right]$$

yields a controller $K = \hat{K}(I + D_{22}\hat{K})^{-1}$ for P. It can be shown that

$$K = \left[\begin{array}{c|c} \hat{A}_k - \hat{B}_k D_{22}(I_{m_2} + \hat{D}_k D_{22})^{-1}\hat{C}_k & \hat{B}_k - \hat{B}_k D_{22}(I_{m_2} + \hat{D}_k D_{22})^{-1}\hat{D}_k \\ \hline (I_{m_2} + \hat{D}_k D_{22})^{-1}\hat{C}_k & (I_{m_2} + \hat{D}_k D_{22})^{-1}\hat{D}_k \end{array} \right].$$

In order to be able to determine K from $\hat{K}$, we must exclude the possibility that the feedback system becomes ill-posed, i.e. $\det(I + \hat{K}(\infty)D_{22}) = 0$.

The following numerical aspects of the presented method should be noted.

1. The main computational task in the Riccati-method is the solution of the X- and Z-Riccati equations ((4) and (5), respectively). The existing software [16] allows to solve such equations in a reliable way up to the order of several hundreds. This software produces estimates of the Riccati equations conditioning and accuracy of solution. But it should be noted that with γ approaching γ_{opt} one or both of the Riccati equations become ill-conditioned. This typically leads to a loss of accuracy in the solution.

2. Apart from the accuracy of the solution of the Riccati equations, the accuracy of the controller matrices depends on the conditioning of the matrices R_3, ∇, S_{t3}, V_{12} and V_{21}. It is not clear what is the connection between the conditioning of these matrices and the conditioning of the original problem. Thus the analysis of the numerical properties of the method is still a mostly open problem.

5 LMI-based $\mathcal{H}_\infty$-optimization

Consider a proper discrete-time plant $P(z)$ with state equations in the form (1) with $D_{22} = 0$ and let $\mathcal{N}_{12}$ and $\mathcal{N}_{21}$ denote the kernels (null spaces) of the concatenated matrices $[B_2^T,\ D_{12}^T]$ and $[C_2,\ D_{21}]$, respectively. The discrete-time suboptimal $\mathcal{H}_\infty$-problem (3) is solvable if and only if there exist two symmetric matrices $R, S \in \mathbb{R}^{n\times n}$, satisfying the following system of three linear matrix inequalities (LMIs) [7]

$$\left[\begin{array}{c|c} \mathcal{N}_{12} & 0 \\ \hline 0 & I \end{array}\right]^T \left[\begin{array}{cc|c} ARA^T - R & ARC_1^T & B_1 \\ C_1RA^T & -\gamma I + C_1RC_1^T & D_{11} \\ \hline B_1^T & D_{11}^T & -\gamma I \end{array}\right] \left[\begin{array}{c|c} \mathcal{N}_{12} & 0 \\ \hline 0 & I \end{array}\right] < 0, \quad (6)$$

$$\left[\begin{array}{c|c} \mathcal{N}_{21} & 0 \\ \hline 0 & I \end{array}\right]^T \left[\begin{array}{cc|c} A^TSA - S & A^TSB_1 & C_1^T \\ B_1^TSA & -\gamma I + B_1^TSB_1 & D_{11}^T \\ \hline C_1 & D_{11} & -\gamma I \end{array}\right] \left[\begin{array}{c|c} \mathcal{N}_{21} & 0 \\ \hline 0 & I \end{array}\right] < 0, \quad (7)$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} > 0. \tag{8}$$

Let (R, S) be any solution of this system of LMIs. Then, a full order γ-suboptimal discrete-time controller $K(z) = C_K(zI - A_K)^{-1}B_K$ is obtained in the following way [6]. One constructs the matrix

$$\Delta = \Delta^T := -\begin{bmatrix} -R & 0 & A + B_2D_KC_2 & B_1 + B_2D_KD_{21} \\ 0 & -\gamma I & C_1 + D_{12}D_KC_2 & D_{11} + D_{12}D_KD_{21} \\ * & * & -S & 0 \\ * & * & 0 & -\gamma I \end{bmatrix} \tag{9}$$

and determines a solution D_K, which satisfies $\Delta > 0$. Then one computes the least-squares solutions $\begin{bmatrix} \Theta_B \\ * \end{bmatrix}$ and $\begin{bmatrix} \Theta_C \\ * \end{bmatrix}$ of the linear systems of equations

$$\left[\begin{array}{c|cccc} 0 & 0 & 0 & C_2 & D_{21} \\ \hline 0 & & & & \\ 0 & & -\Delta & & \\ C_2^T & & & & \\ D_{21}^T & & & & \end{array}\right] \left[\begin{array}{c} \Theta_B \\ \hline \\ \\ * \\ \\ \end{array}\right] = -\left[\begin{array}{c} 0 \\ \hline -I \\ 0 \\ A^TS \\ B_1^TS \end{array}\right], \tag{10}$$

$$
\left[\begin{array}{c|cccc} 0 & B_2^T & D_{12}^T & 0 & 0 \\ \hline B_2 & & & & \\ D_{12} & & -\Delta & & \\ 0 & & & & \\ 0 & & & & \end{array}\right]
\left[\begin{array}{c} \Theta_C \\ \hline \\ \\ * \\ \\ \end{array}\right] = -\left[\begin{array}{c} 0 \\ \hline AR \\ C_1 R \\ -I \\ 0 \end{array}\right]. \tag{11}
$$

Then one computes, via the computation of singular value decompositions (SVDs), two invertible matrices $M, N \in \mathbb{R}^{n\times n}$ such that

$$
MN^T = I - RS, \tag{12}
$$

and then matrices A_K, B_K and C_K by solving

$$
\begin{aligned}
NB_K &= -SB_2D_K + \Theta_B^T, && (13)\\
C_K M^T &= -D_K C_2 R + \Theta_C, && (14)\\
-NA_K M^T &= SB_2\Theta_C + \Theta_B^T C_2 R + S(A - B_2 D_K C_2)R \\
&\quad + \begin{bmatrix} -I \\ 0 \\ A^T S + C_2^T \Theta_B \\ B_1^T + D_{21}^T \Theta_B \end{bmatrix}^T \Delta^{-1} \begin{bmatrix} AR + B_2\Theta_C \\ C_1 R + D_{12}\Theta_C \\ -I \\ 0 \end{bmatrix}. && (15)
\end{aligned}
$$

Note that these formulas for controller matrices may be used also in the case of the Riccati-based approach by setting $R^{-1} = \gamma^{-1}X$ and $S^{-1} = \gamma^{-1}Y$, where X and Y are the solutions of the corresponding Riccati equations.

The *LMI-method* has the following numerical properties.

1. Computing solutions (R, S) of the LMI system (6)-(8) is a convex optimization problem. The available polynomial type algorithms [3, 13] allow to solve this LMI problem with a complexity of $O(n^6)$ operations. Also, it should be noted that the sensitivity of the LMIs under consideration, subject to variations in the plant data, may affect the accuracy of the matrices R and S and hence the accuracy of controller matrices. Unfortunately, it is an open problem how the sensitivity of the LMIs is connected to the sensitivity of the given $\mathcal{H}_\infty$-suboptimal problem.

2. The systems of equations (10) and (11) are ill-conditioned, when the matrix Δ is nearly singular. In such a case, the solutions Θ_B and Θ_C are of large norms, which results in fast controller dynamics and numerical instability when closing the loop. This difficulty may be removed by increasing the value of γ.

3. The matrices M and N are ill-conditioned when $I - RS$ is nearly singular. This may affect the accuracy of the solution of the equations (13)-(15). As shown in [7], the rank of the matrix $I - RS$ determines the controller order. If this matrix has a numerical rank less than n then it is possible to switch to formulas for reduced order controllers.

On the basis of these brief discussions, it is difficult to make a firm conclusion about the numerical properties of the presented methods. However, some impression about the numerical behavior of these methods may be derived from the numerical experiments presented below.

6 Available software

In our experiments we compare the performance of the following routines for the solution of the discrete-time $\mathcal{H}_\infty$-optimization problem.

- `dhfsyn` Routine from the μ-toolbox [1] utilizing a bilinear transformation to a continuous-time system. The user has to supply a tolerance, which determines the relative difference between final γ-values in the bisection procedure.

- `dhinf` Routine from Robust Control Toolbox [4] also implementing a bilinear transformation to a continuous-time system. There is no tolerance set by the user.

- `dhinfric` Routine from the LMI toolbox [9] based on the solution of two discrete-time algebraic Riccati equations. The controller matrices are determined by the formulas given in Section 5 with $R^{-1} = \gamma^{-1}X$ and $S^{-1} = \gamma^{-1}Y$. Tolerance determining the required relative accuracy in finding γ_{opt} may be set by the user.

- `dhinflmi` Routine from the LMI toolbox [9], implementing the LMI-based method described in Section 5. The routine has the capability to reduce the controller order when possible. A tolerance equal to the required relative accuracy may be set by the user.

- `dhw` M-file intended for the design of a suboptimal $\mathcal{H}_\infty$-controller implementing the method described in Section 4. There is a provision to compute the exact condition numbers of the Riccati equations involved.

- `dishin` mex-file from the SLICOT library utilizing the Fortran 77 routine `SB10DD`. It implements the same formulae as `dhw`, except that instead of the exact condition numbers of the Riccati equations one computes their cheap estimates.

In the Appendix we give the M-file `dhw` along with the M-file `cndricd` intended for determination of the exact condition numbers of the corresponding Riccati equations. It should be noted that in the most recent version of MATLAB, some of these routines are not availbale anymore.

7 Numerical experiments

First we present a simple example illustrating the difficulties arising in a particular case of $\mathcal{H}_\infty$-optimization design.

Example 1 Consider the discrete-time system

$$\begin{aligned} x_{k+1} &= Ax_k + B_1 w_k + B_2 u_k, \\ z_k &= C_1 x_k + D_{11} w_k + D_{12} u_k, \\ y_k &= C_2 x_k + D_{21} w_k + D_{22} u_k \end{aligned}$$

where $x_k \in \mathbb{R}^3,\ w_k \in \mathbb{R}^1,\ u_k \in \mathbb{R}^1,\ y_k \in \mathbb{R}^1,\ zk \in \mathbb{R}^1$ and

$$A = \begin{bmatrix} -2.1 & -1 & 1 \\ 4 & -1.8 & 2 \\ -2 & 1 & -0.7 \end{bmatrix},\ B_1 = B_2 = \begin{bmatrix} -1 \\ -2 \\ 1 \end{bmatrix},$$

$$C_1 = C_2 = \begin{bmatrix} -2 & 1 & -1 \end{bmatrix},\ D_{11} = 1,\ D_{12} = 1,\ D_{21} = 1,\ D_{22} = 0.$$

This example is interesting due to the fact that for the given system it is possible to use a controller which compensates entirely the effect of the disturbance so that the transfer function between the disturbance w_k and the controlled output z_k becomes equal to 0. In this way the optimum value of the $\mathcal{H}_\infty$-norm is equal to 0 and the optimal controller reduces to the static output feedback $K = -1$. For this controller the equations of the closed-loop system become

$$\begin{aligned} x_{k+1} &= (A - B_2 C_2) x_k + (B_1 - B_2 D_{21}) w_k, \\ z_k &= (C_1 - D_{12} C_2) x_k + (D_{11} - D_{12} D_{21}) w_k, \\ y_k &= -u_k \end{aligned}$$

For the chosen matrices we have

$$B_1 - B_2 D_{21} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$C_1 - D_{12} C_2 = [0 \;\; 0 \;\; 0]\,,$$

$$D_{11} - D_{12} D_{21} = 0$$

and the closed-loop system with state matrix

$$A - B_2 C_2 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$$

is asymptotically stable.

This example represents a difficult test for some of the routines under consideration as shown below.

DHFSYN

This routine was not able to produce a solution until *tol* was increased to $10^8 \times$ `eps` where `eps` is the relative machine precision. After several warnings the routine produced the following result.

```
gfin =

   1.4901e-08


ak =

    0.1000   -0.0000    0.0000
   -0.0000    0.2000   -0.0000
   -0.0000    0.0000    0.3000


bk =

   1.0e-14 *

    0.3544
   -0.0000
   -0.1510
```

```
ck =

   1.0e-14 *

    0.6112   -0.0000    0.0725

dk =

   -1.0000
```

where $gfin$ is the computed approximation of γ_{opt}.

Note that the controller is stable, thus the internal stability of the closed-loop system is ensured.

DHINF

This routine produces accurate result for the example under consideration.

```
ak =

    0.1000   -0.0000    0.0000
    0.0000    0.2000    0.0000
   -0.0000    0.0000    0.3000

bk =

     0
     0
     0

ck =

     0     0     0

dk =

    -1
```

DHINFLMI

This routines needed 25 iterations in order to find the optimal performance

```
gfin =

  2.3563e-012
```

with tolerance equal to `eps` and guaranteed absolute accuracy 9.5×10^{-11}. The controller obtained is

```
ak =

     []

bk =

   Empty matrix: 0-by-1

ck =

   Empty matrix: 1-by-0

dk =

   -1.0000
```

which is the correct answer of the problem.

DHINFRIC

This routine required a large tolerance in order to produce a solution. For $tol = 5 \times 10^{11}$ `eps` the following result was obtained.

```
gfin =

    0.0033

ak =

    0.1000   -0.0000    0.0000
    0.0000    0.3000   -0.0016
   -0.0000   -0.0016    0.2000
```

```
bk =

  1.0e-005 *

   -0.4581
    0.4432
   -0.9236

ck =

  1.0e-005 *

    0.9163    0.4506   -0.4656

dk =

   -1.0000
```

DHW

In this case we were able to produce a solution for $\gamma = 10^8 \times$ `eps`. The following controller matrices were obtained.

```
ak =

    0.1000         0         0
    0.0000    0.2000    0.0000
   -0.0000    0.0000    0.3000

bk =

  1.0e-015 *

         0
    0.4441
   -0.2220

ck =
```

```
     0      0      0

dk =

    -1
```

The reciprocal condition numbers of the corresponding Riccati equations are

```
xcond =

    0.1447

ycond =

    0.1852
```

so that the Riccati equations are very well conditioned.

DISHIN

This routine also required a large tolerance $tol = 10^{11} \times$ `eps` in order to produce a solution. The controller matrices obtained are

```
ak =

    0.1000   -0.0000    0.0000
   -0.0000    0.2000    0.0000
    0.0000   -0.0000    0.3000

bk =

  1.0e-014 *

   -0.0444
   -0.3109
    0.2220

ck =

  1.0e-014 *

   -0.1776    0.1776   -0.3775
```

```
dk =

   -1.0000
```

In this way only the routine DHINFLMI was able to remove the unnecessary dynamics of the controller producing static output feedback. This routine also found the smallest value for the optimal $\mathcal{H}_\infty$-norm. The most inaccurate results for this particular case were produced by the routine DHINFRICC.

Example 2 Consider a family of discrete-time $\mathcal{H}_\infty$- optimization problems for a system with

$$A = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1+\alpha \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & -0.1 & 0 \\ 0 & 0 & 0.1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$D_{11} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}, \quad D_{12} = \begin{bmatrix} 0 & 0 \\ \beta & 0 \\ 0 & \beta \end{bmatrix},$$

$$D_{21} = \begin{bmatrix} 0 & \eta & 0 \\ 0 & 0 & \eta \end{bmatrix}, \quad D_{22} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Note that the pair (A, B_2) is stabilizable and the pair (A, C_2) is detectable. When α is decreased, then two open-loop poles approach the unit circle. This creates difficulties for the methods based on the solution of Riccati equations. If the parameter β (η) is zero then the condition **A2** (**A3**), given in Section 4, is violated, i.e. the plant becomes singular. Changing parameters α, β, η allows to reveal the numerical properties of the different methods for $\mathcal{H}_\infty$-optimization.

7.1 Influence of α

Consider first the influence of the parameter α on the design. For this aim we vary α between 10^{-12} and 5.37×10^{-4} keeping $\beta = 1$ and $\eta = 1$.

DHFSYN and **DHINF**

These routines were not able to produce a solution for α in this region, apparently due to numerical difficulties associated with the used bilinear transformation.

DHINFLMI

This routine is used with a tolerance for the relative accuracy in determining the optimal $\mathcal{H}_\infty$-norm equal to the default value (10^{-2}). In Table 7.1 we show the number of iterations in solving the LMIs as a function of α. The number of iterations has a maximum for $\alpha = 3.28 \times 10^{-8}$.

The optimum value of γ as a function of α is given in Figure 1. There is a sudden decrease of γ_{opt} for α between 10^{-8} and 10^{-7}.

In Figure 2 we show the dependence of the norms of the LMI solutions R and S as functions of α. These norms increase slightly when a change in α is causing a large change of γ_{opt}.

As we can see from Figure 3, the norms of the controller matrices do not depend on α.

In this way we see that the change of the parameter α does not significantly influence the solution of the $\mathcal{H}_\infty$-problem found by the LMI-based method. We will see below, however, that the variation of this parameter profoundly affects the methods based on the solution of Riccati equations.

DHINFRIC

This routine is used with the defaults tolerance 10^{-2} for the relative accuracy of the optimal $\mathcal{H}_\infty$-norm a(10^{-2}).

Table 1: Number of iterations as a function of α

k	α	Number of iterations
1	1×10^{-12}	16
2	2×10^{-12}	16
3	4×10^{-12}	16
4	8×10^{-12}	16
15	1.64×10^{-12}	23
16	3.28×10^{-8}	24
17	6.56×10^{-8}	21
18	1.31×10^{-7}	20
19	2.62×10^{-7}	19
20	5.24×10^{-7}	19
21	1.05×10^{-6}	18
22	2.09×10^{-6}	17
23	4.19×10^{-6}	15
24	8.39×10^{-6}	16
25	1.68×10^{-5}	17
26	3.56×10^{-5}	16
27	6.71×10^{-5}	17
28	1.34×10^{-4}	17
29	2.68×10^{-4}	16
30	5.37×10^{-4}	15

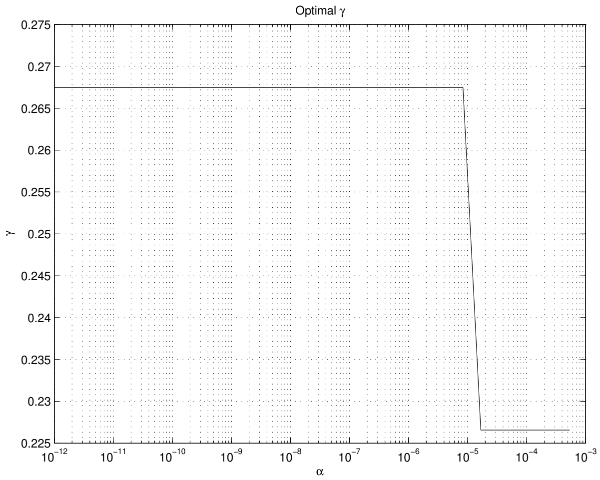

Figure 1: **DHINFLMI:** Optimal γ as a function of α

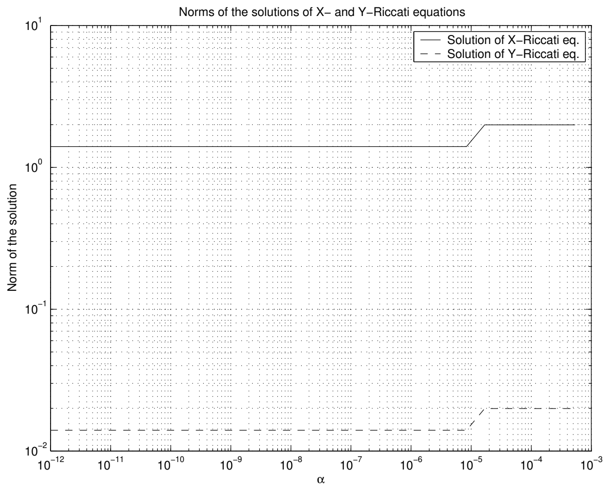

Figure 2: **DHINFLMI:** Norms of R and S as functions of α

The norms of the Riccati solutions X and Y as function of α is depicted in Figure 4. They have the same behavior as the norms of the LMI solutions in Figure 2.

The norms of the controller matrices (Figure 5) remain constant when

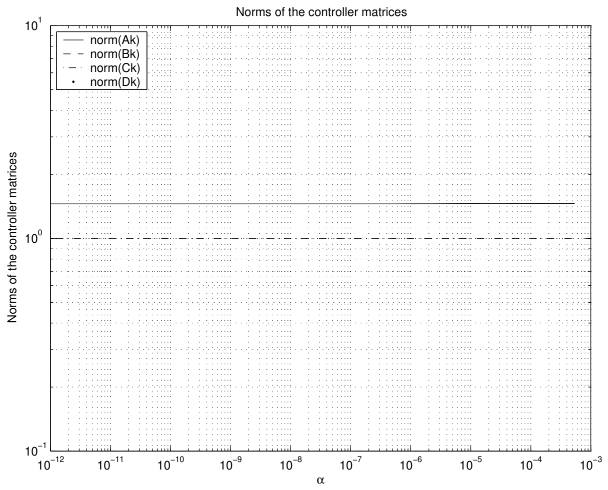

Figure 3: **DHINFLMI:** Norms of controller matrices as functions of α

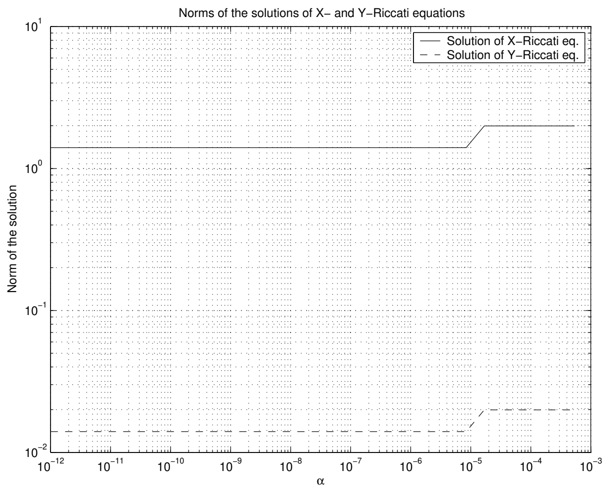

Figure 4: **DHINFRIC:** Norms of X and Y as functions of α

α is varied.

This routine does not produce any information about the conditioning of the two Riccati equations that have to be solved. The next two routines provide such information and it is seen that with the decrease of α the

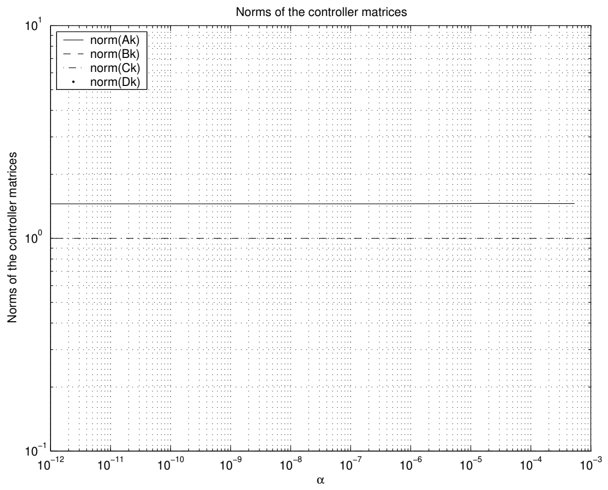

Figure 5: **DHINFRIC:** Norms of controller matrices as functions of α

conditioning of the Riccati equations may deteriorate.

DHW

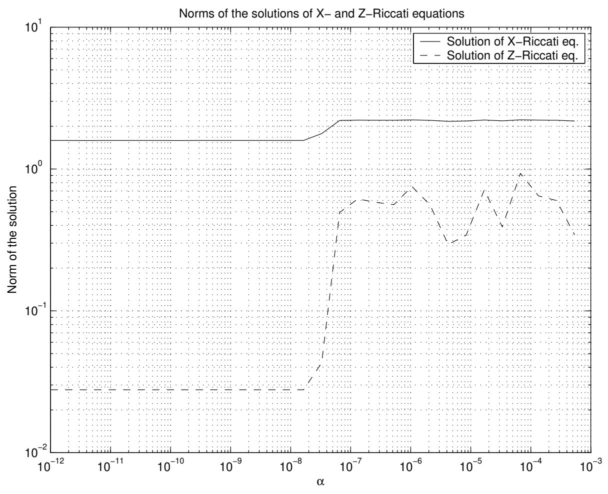

Figure 6: **DHW:** Norms of X and Z as functions of α

The norms of the Riccati solutions X and Z are depicted in Figure

6, which shows that the norm of the solution Z varies non-smoothly with increasing α.

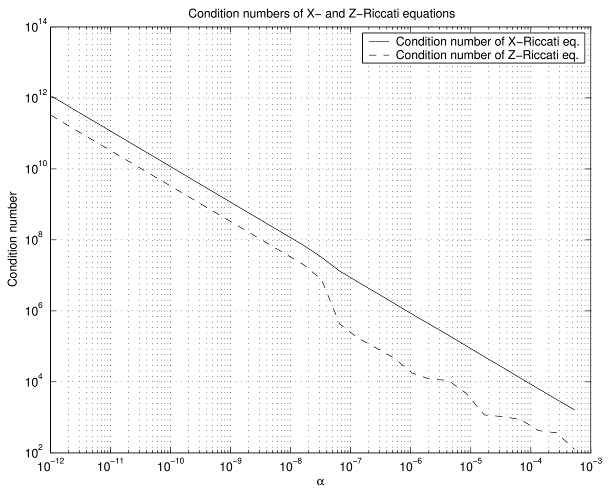

Figure 7: **DHW:** Exact condition numbers of the Riccati equations as functions of α

In Figure 7 we see that the conditioning of both Riccati equations increases almost inversely proportional to α. Hence for values of α approaching the machine precision from above, the solutions of the Riccati equations may be entirely wrong. In this way the two open-loop poles, approaching the unit circle in the complex plane, have a significant influence on the accuracy of solution. This effect may be considered entirely as a result of the reformulation of the $\mathcal{H}_\infty$-optimization problem as optimization problem based on Riccati equations. It is interesting to note that the controller matrices again have constant norms, as in the case when the LMI formulation of the problem is used (Figure 8).

In Figures 9 and 10 we show the behavior of the Riccati solutions and the conditioning of the Riccati equations, respectively, for α varying in the range $10^{-8} - 10^{-7}$. We see that in this range the solution of the Z-equation and the conditioning of this equation are very sensitive to changes in α.

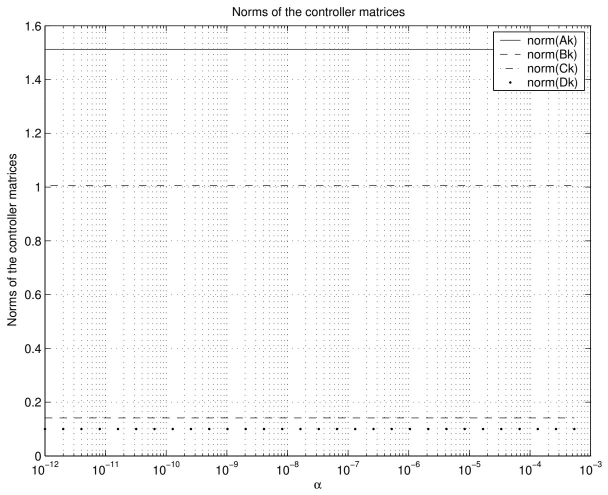

Figure 8: **DHW:** Norms of controller matrices as functions of α

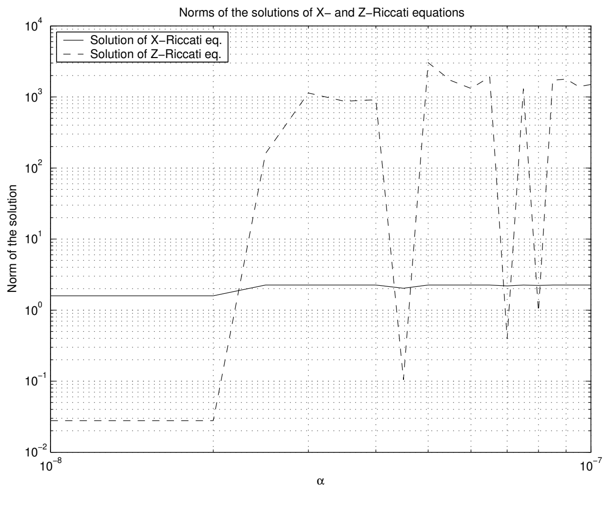

Figure 9: **DHW:** Norms of X and Z as functions of α

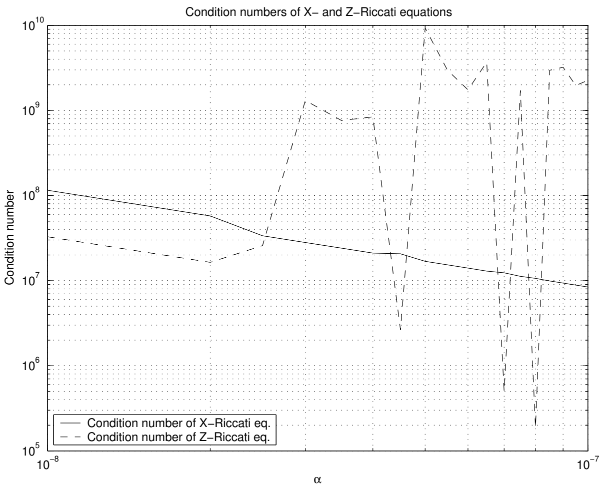

Figure 10: **DHW:** Exact condition numbers of the Riccati equations as functions of α

DISHIN

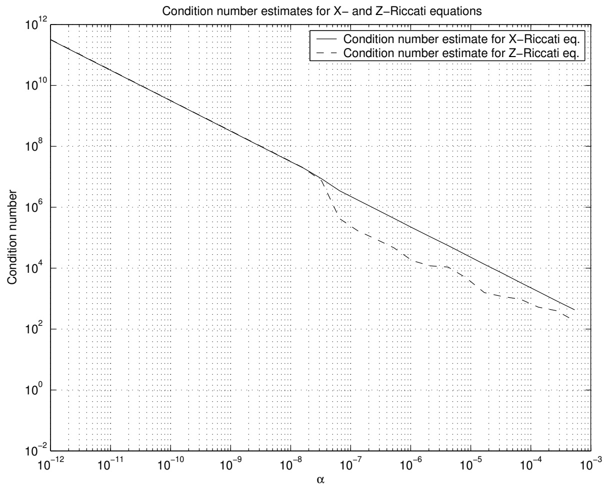

Figure 11: **DISHIN:** Estimates of the condition number of the Riccati equations as function of α

The results from this routine coincide with the results from routine **DHW** as one may expect. That is why in Figure 11 we show only the estimates of the condition numbers of the solved Riccati equations. We see that the routine produces underestimates which represent correctly the behavior of the true condition numbers.

7.2 Influence of β

The values of the parameters β and η may also have a significant effect on the solution of the $\mathcal{H}_\infty$-optimization problem for the example under consideration. In what follows we consider the results obtained by the different routines. In these experiments we take $\alpha = 10^{-4}$ and $\eta = 1$.

DHINFLMI

In Table 7.2 we give the number of iterations in solving the LMIs for different values of β. The maximum number of iterations is achieved for $\beta = 3.2 \times 10^{-5}$.

The optimum values of γ for different values of β are shown in Figure 12. With β approaching zero, γ is tending to a constant value.

Table 2: Number of iterations as a function of β

k	β	Number of iterations
1	1×10^{-6}	35
2	2×10^{-6}	35
3	4×10^{-6}	35
4	8×10^{-6}	49
5	1.6×10^{-5}	53
6	3.2×10^{-5}	61
7	6.4×10^{-5}	37
8	1.28×10^{-4}	37
9	2.56×10^{-4}	49
10	5.12×10^{-3}	41

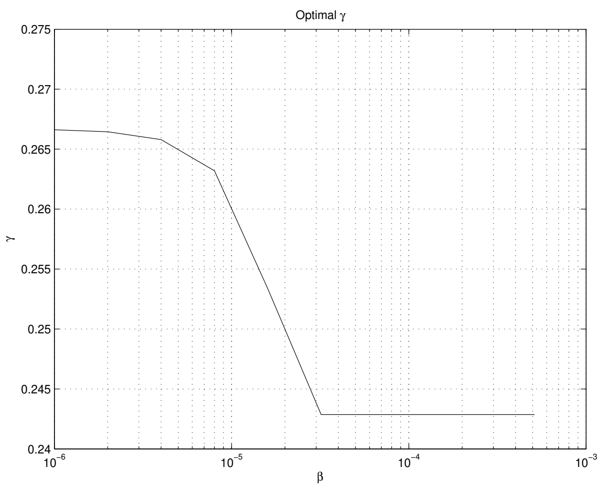

Figure 12: **DHINFLMI:** Optimal γ as a function of β

In Figure 13 we show the dependence of the norms of the LMI solutions on β and in Figure 14 we show the norms of the controller matrices as functions of this parameter.

In order to investigate more carefully the behavior of **DHINFLMI**, we performed similar experiments for $\alpha = 0,\ \eta = 0$ and β varying in the range

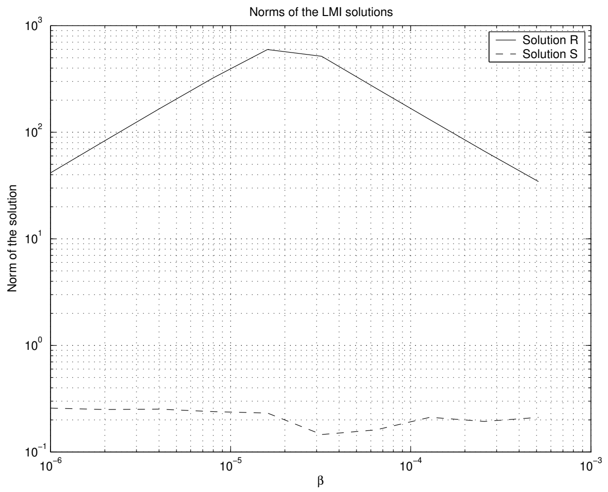

Figure 13: **DHINFLMI:** Norms of R and S as functions of β

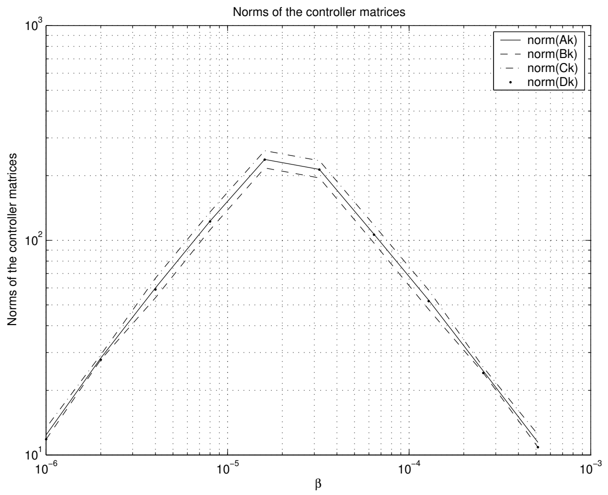

Figure 14: **DHINFLMI:** Norms of controller matrices as functions of β

$10^{-14} - 2^{10} \times 10^{-14}$. (Note that for these values of α and η the other routines are not able to produce solutions.)

We see from Figure 15 that the optimum value of γ remains constant even for values of β approaching the machine precision.

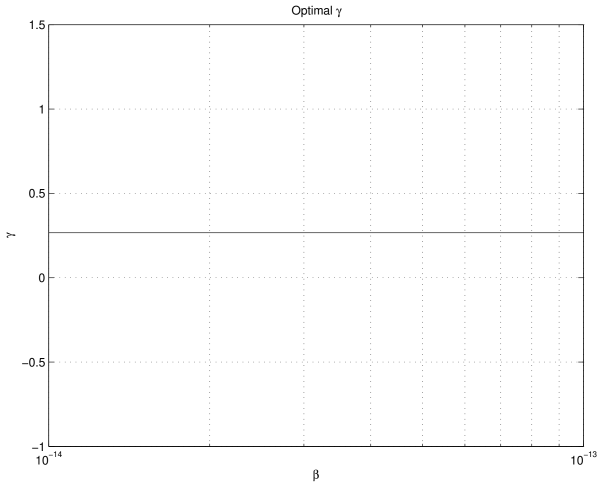

Figure 15: **DHINFLMI:** Optimal γ for small values of β

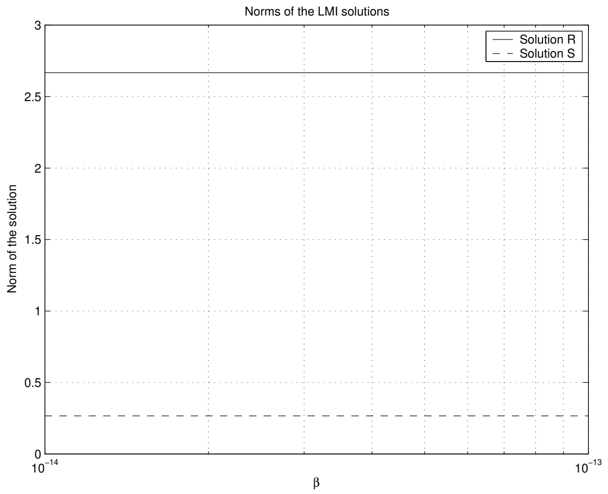

Figure 16: **DHINFLMI:** Norms of R and S for small values of β

The norms of LMIs solutions and the norms of controller matrices are shown in Figures 16 and 17, respectively.

It is interesting also to investigate the behavior of **DHINFLMI** for different values of the tolerance *tol* used in the determination of the optimal

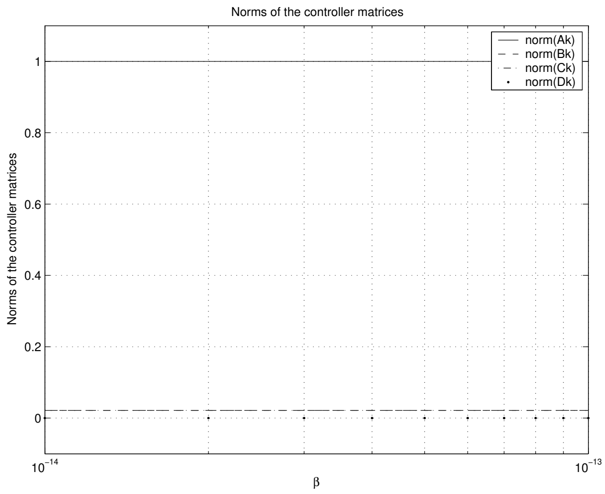

Figure 17: **DHINFLMI:** Norms of controller matrices for small values of β

γ.

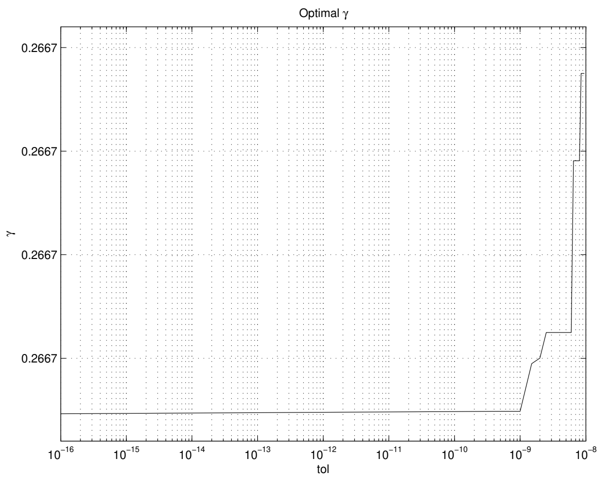

Figure 18: **DHINFLMI:** Optimal γ for different values of *tol*

In Figures 18 and 19 we show the dependence of the optimal value of γ and the norms of the LMIs solutions, respectively, on *tol*. Note that γ and $\|R\|$, $\|S\|$ decrease smoothly with *tol* even for a tolerance tending to the

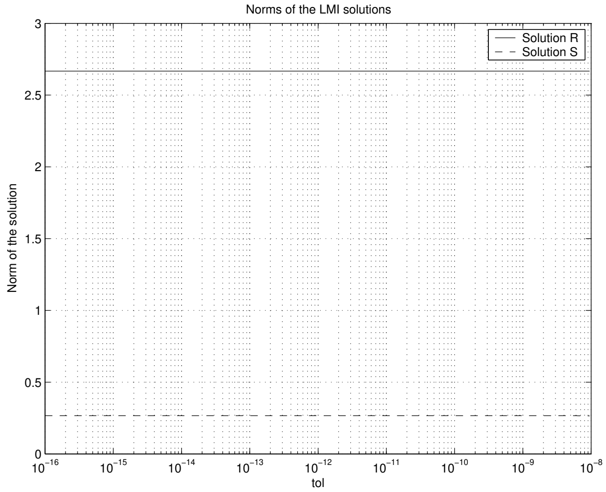

Figure 19: **DHINFLMI:** Norms of R and S for different values of *tol*

machine precision.

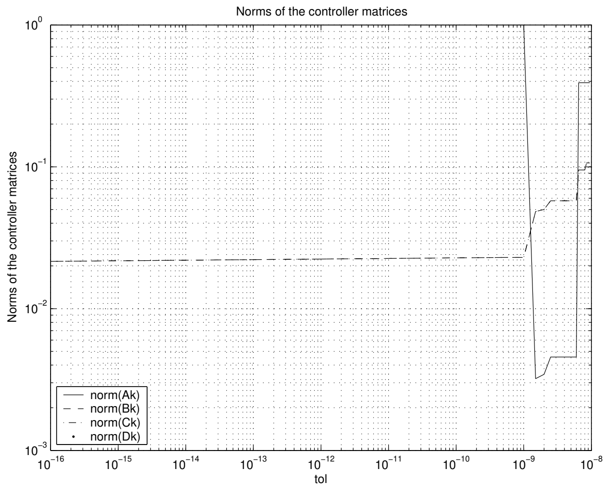

Figure 20: **DHINFLMI:** Norms of controller matrices for different values of *tol*

The norms of the controller matrices, however, change significantly with *tol* increasing to 10^{-9} as seen from Figure 20. This is explained by the

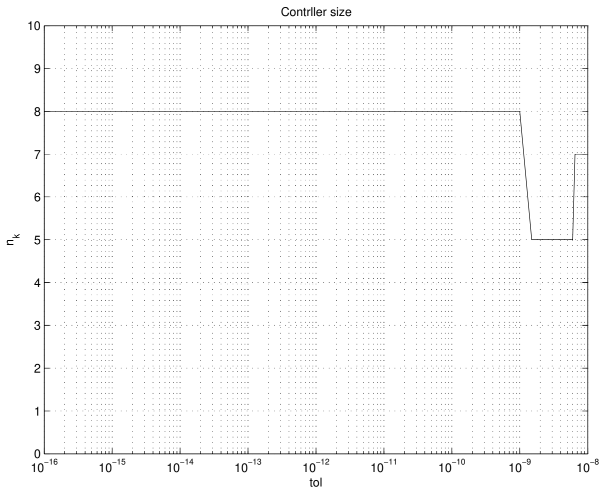

Figure 21: **DHINFLMI:** Size of the controller for different values of *tol*

change of the controller size for this value of *tol* as shown in Figure 21. This is a useful capability of the routine **DHINFLMI** which allows to reduce the size of the controller whenever possible.

DHINFRIC

The routine **DHINFRIC** was unable to solve the discrete-time $\mathcal{H}_\infty$-optimization problem for values of β smaller than 10^{-7}. We see from Figure 22 that for β decreasing to 10^{-6} the values of γ increase to 0.29 which is larger than the value obtained by **DHINFLMI** (0.2667).

The norms of the Riccati solutions tend to decrease with the decrease of γ. (Figure 23.)

Figure 24 reveals a disadvantage of **DHINFRIC** which is typical for the routines based on Riccati equations. The norms of the controller matrices increase inversely proportional with β and η (inversely proportional with the minimum singular value of D_{12} or D_{21} in the general case).

In Figure 25 we show the norms of the Riccati solutions for small values of β. It is interesting that the norms do not change smoothly when varying β.

For β tending to 10^{-7} the norms of the controller matrices approach 10^7 (Figure 26) which is unacceptable from the point of view of controller implementation.

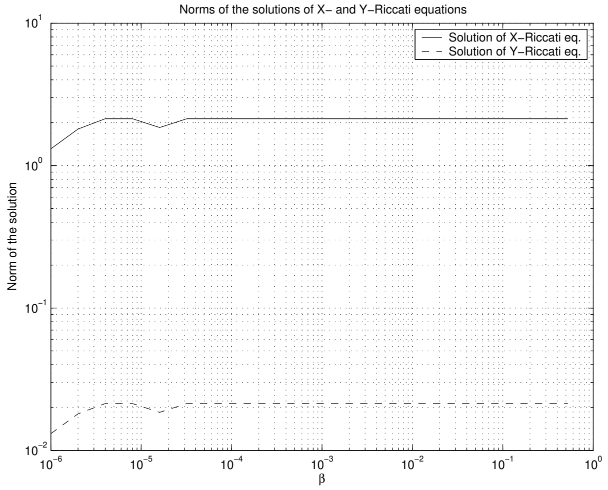

Figure 22: **DHINFRIC:** Optimal γ as a function of β

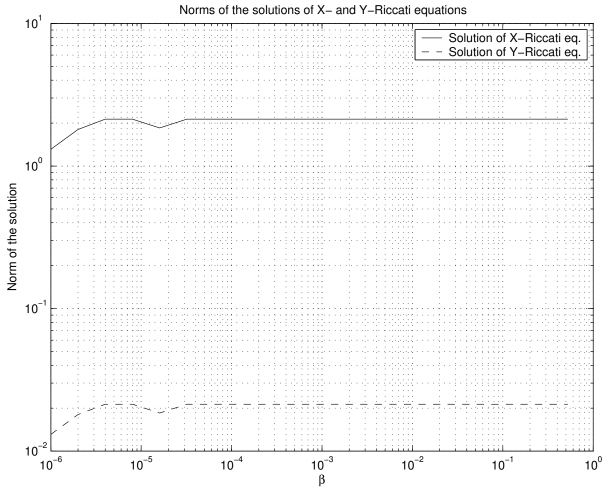

Figure 23: **DHINFRIC:** Norms of X and Y as functions of β

DHW

The norms of the Riccati solutions X and Z as functions of β are depicted in Figure 27. Although the routine **DHW** is also based on the two Riccati equations solutions as the routine **DHINFRIC** we see a different behavior

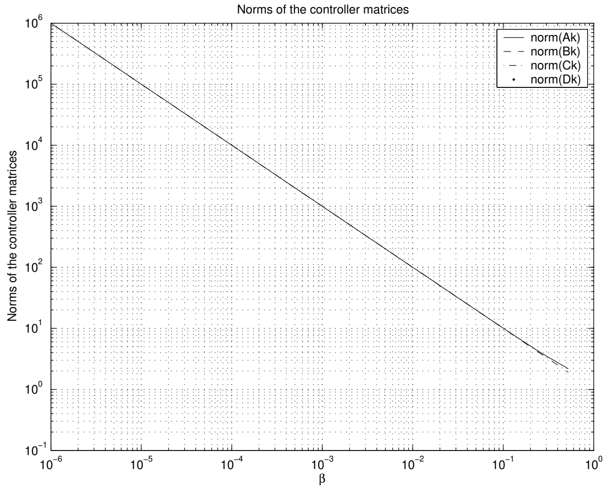

Figure 24: **DHINFRIC:** Norms of controller matrices as functions of β

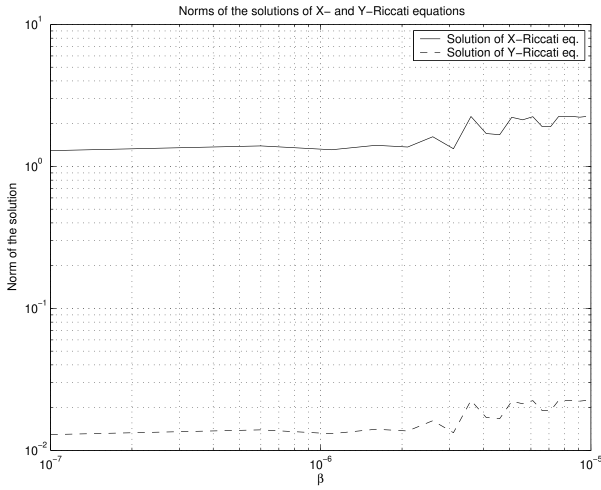

Figure 25: **DHINFRIC:** Norms of X and Y as functions of β

of these solutions (compare with Figure 23).

We see from Figure 28 that the condition number of the X-Riccati equations approaches 10^{16} when β decreases to 10^{-6}. This explains the fact that the routine **DHW** was unable to find a solution for $\beta < 10^{-6}$.

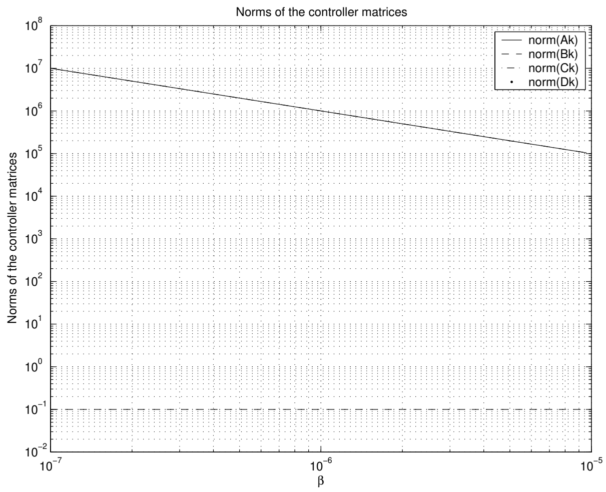

Figure 26: **DHINFRIC:** Norms of controller matrices as functions of β

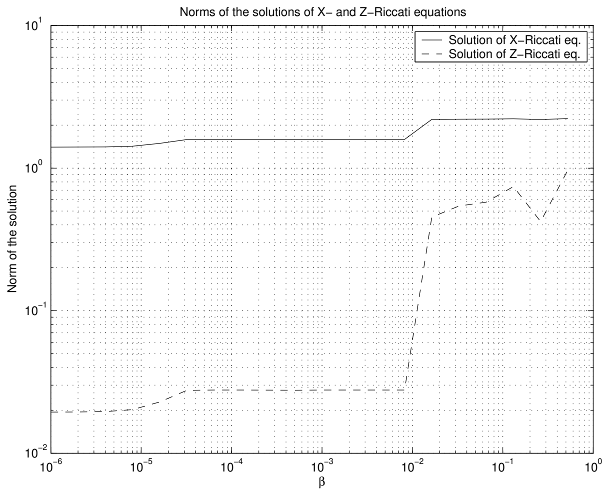

Figure 27: **DHW:** Norms of X and Z as functions of β

The norms of the controller matrices increase with the decrease of β exactly as in the case of **DHINFRIC**.

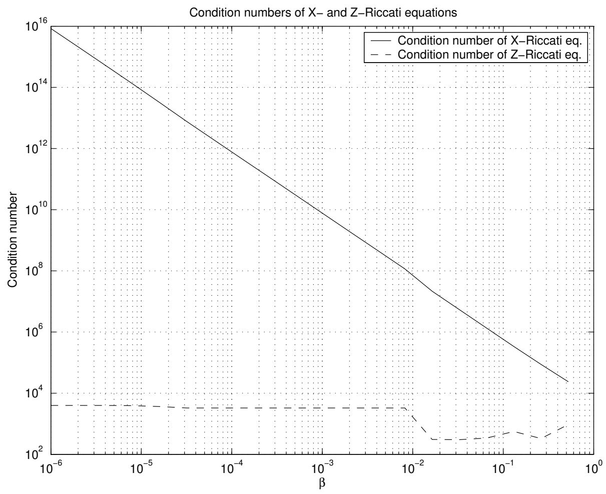

Figure 28: **DHW:** Exact condition numbers of the Riccati equations as functions of β

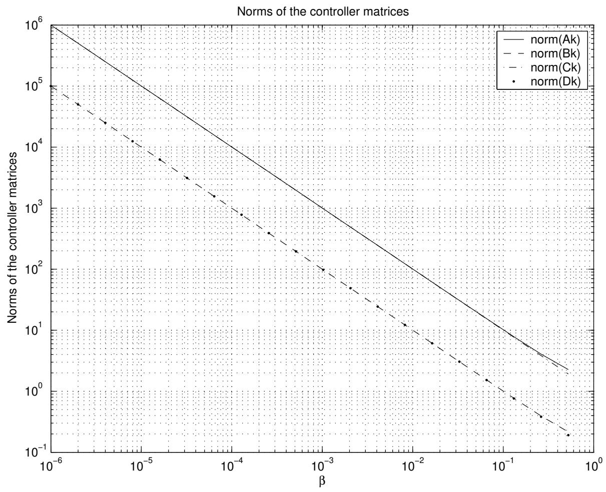

Figure 29: **DHW:** Norms of controller matrices as functions of β

DISHIN

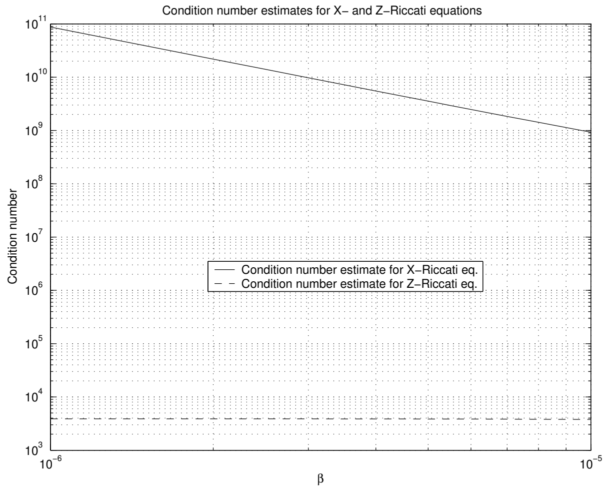

Figure 30: **DISHIN:** Estimates of the condition number of the Riccati equations as function of β

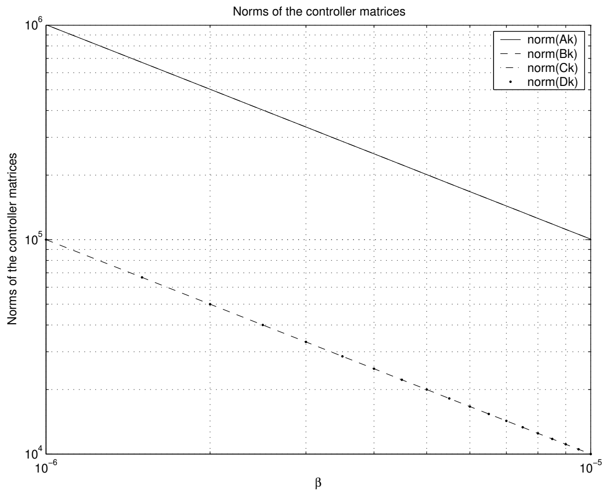

Figure 31: **DISHIN:** Norms of controller matrices as functions of β

The behavior of this routine with the variation of β is similar to this of

DHW. **DISHIN** produces a solution for $\beta > 10^{-6}$ and the conditioning of the X-Riccati equation and the controller matrices norms increase rapidly with the decrease of β (Figures 30 and 31).

8 Conclusions

The following conclusions can be derived from the restricted number of numerical experiments done.

- The experiments show serious disadvantages of the routines based on Riccati equations. In the cases when an invariant zero of P_{12} or P_{21} tends to the unit circle, the condition numbers of the Riccati equations approach 1/`eps` while in the same cases the solution obtained by the LMI-based method does not show any singularity and allows to obtain smaller values of γ_{opt}. The norms of the controller matrices obtained by the Riccati approach become unacceptably large when the matrices D_{12} or D_{21} tend to matrices with smaller numerical rank.

- It is clear that the fulfillment of conditions **A2** and **A3** (Section 4) is important only for the methods based on Riccati equations. The violation of these conditions does not affect the method based on LMIs. This confirms that the terms 'regular plant' and 'singular plant' are meaningful only in the context of the Riccati-based methods.

- Although the LMI based method produces the best solution it is necessary to take into account that the numerical properties of the second step of this method - the computation of controller matrices - are not studied well. A numerical analysis showing some type of stability of the method is still needed.

- It seems appropriate to study the sensitivity of the original discrete-time $\mathcal{H}_\infty$-optimization problem and the sensitivity of the corresponding LMI formulations.

- High performance software for solving $\mathcal{H}_\infty$-optimization problems should include methods for regularization of ill-conditioned problems. Such method have to provide automatically the provision to use controllers of different size. It seems justified to combine in some way the Riccati-based approach and the LMI approach in order to develop an efficient and numerically reliable $\mathcal{H}_\infty$-optimization method.

Acknowledgments

Example 1 was inspired by discussions with Ralph Byers.

The second author is grateful to Deutscher Akademischer Austauch Dienst for supporting his visit to TU-Berlin.

References

[1] G.J. Balas, J.C. Doyle, K. Glover, A. Packard, R. Smith. *μ-Analysis and Synthesis Toolbox*. The MathWorks, Inc.,Natick, MA, 1998.

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

[3] S. Boyd, L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[4] R.Y. Chiang, M.G. Safonov. *Robust Control Toolbox*. The MathWorks, Inc.,Natick, MA, 1996.

[5] J.C. Doyle, K. Glover, P. Khargonekar, B. Francis. State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. *IEEE Trans. Automat. Control*, **AC-34**, pp. 831-847.

[6] P. Gahinet. Explicit controller formulas for LMI-based $\mathcal{H}_\infty$ synthesis. *Automatica*, **32**, pp. 1007-1014, 1996.

[7] P. Gahinet, P. Apkarian. A linear matrix inequality approach to $\mathcal{H}_\infty$ control. *Int. J. Robust Nonlinear Control*, **54**, pp. 421-448, 1994.

[8] P. Gahinet, A.J. Laub. Numerically reliable computation of optimal performance in singular $\mathcal{H}_\infty$ control. *SIAM J. Control Optim.*, **35**, pp. 1690-1710, 1997.

[9] P. Gahinet, A. Nemirovski, A.J. Laub, M. Chilali. *LMI Control Toolbox*. The MathWorks, Inc.,Natick, MA, 2000.

[10] M. Green, D.J.N. Limebeer. *Linear Robust Control*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[11] T. Kailath. *Linear Systems*. Prentice Hall, Englewood Cliffs, NJ, 1980.

[12] P. Iglesias, K. Glover. State-space approach to discrete-time $\mathcal{H}_\infty$-control. *International J. of Control*, **54**, pp. 1031-1074, 1991.

[13] Yu. Nesterov, A. Nemirovski. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. SIAM, Philadelphia, 1994.

[14] P.Hr. Petkov, D.-W. Gu, M.M. Konstantinov. Fortran 77 Routines for $\mathcal{H}_\infty$ and $\mathcal{H}_2$ Design of Discrete-Time Linear Control Systems. *NICONET Report 1995*. Available electronically at http:www.win.tue.nl/NICONET

[15] M.G. Safonov, D.J.N. Limebeer, R.Y. Chiang. Simplifying the $\mathcal{H}_\infty$ theory via loop shifting, matrix pencil and descriptor concepts. *International J. of Control*, **50**, pp. 2467-2488, 1989.

[16] SLICOT library. Available electronically at http:www.win.tue.nl/NICONET

[17] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River, NJ, 1995.

Appendix. M-files for discrete-time $\mathcal{H}_\infty$ optimization and estimation of Riccati equation conditioning

```
function [al,bl,cl,dl,ak,bk,ck,dk,x,y,xcond,ycond] = dhw(a,b,c,d,m2,p2,gamma)
%
% Design of a discrete-time H-infinity suboptimal controller
% implementing the formulas from Green and Limebeer
%
[n,m] = size(b);
[p,n] = size(c);
m1 = m - m2;
p1 = p - p2;
%
b1 = b(:,1:m1);
b2 = b(:,m1+1:m);
c1 = c(1:p1,:);
c2 = c(p1+1:p,:);
%
d11 = d(1:p1,1:m1);
d12 = d(1:p1,m1+1:m);
d21 = d(p1+1:p,1:m1);
d22 = d(p1+1:p,m1+1:m);
%
d1d = [d11 d12];
r = d1d'*d1d - [gamma^2*eye(m1)  zeros(m1,m2)
                 zeros(m2,m1)    zeros(m2,m2)];
ax = a - b*(r\d1d')*c1;
cx = c1'*c1 - c1'*d1d*(r\d1d')*c1;
%if norm(cx)<sqrt(eps)
%   cx = zeros(n);
%end
%
% Solution of X-Riccati equation
%
%cx
%[x,l,g,rr] = dare(ax,b,cx,r);
[x,l,g,rr] = dare(a,b,c1'*c1,r,c1'*d1d);
%[x,w,rcond,ferr] = ricdsolv(1,ax,cx,b*(r\b'));
%[rcon,ferr] = drcon(a,b*(r\b'),c1'*c1,x);
```

```
xcond = cndricd(ax,cx,b*(r\b'),x);
%rcon
%ex = eig(x)
%
% Accuracy test
%
%resx = x-a'*x*a-c1'*c1+(a'*x*b+c1'*d1d)*((r+b'*x*b)\(a'*x*b+c1'*d1d)');
%norm_resx = norm(resx)/norm(x)
%
r1 = d11'*d11 - (gamma^2)*eye(m1)+b1'*x*b1;
r2 = d12'*d11 + b2'*x*b1;
r3 = d12'*d12 + b2'*x*b2;
%con_r3 = cond(r3)
%r = [r1 r2';r2 r3];
h1 = d11'*c1 + b1'*x*a;
h2 = d12'*c1 + b2'*x*a;
h = [h1;h2];
%eig_l = eig(a-b*(r\h))
del = r1 - r2'*(r3\r2);
%con_del = cond(del)
%eig_delta = eig(del)
ldel = h1 - r2'*(r3\h2);
v12 = chol(r3);
%con_v12 = cond(v12)
v21 = chol(-del/gamma^2);
%con_v21 = cond(v21)
%
at = a - b1*(del\ldel);
bt1 = b1/v21;
ct1 = v12*(r3\h2 - r3\r2*(del\ldel));
ct2 = c2 - d21*(del\ldel);
ct = [ct1
      ct2];
dt11 = v12*(r3\r2)/v21;
dt21 = d21/v21;
dd1 = [dt11
       dt21];
%
rt = dd1*dd1' - [gamma^2*eye(m2) zeros(m2,p2)
                 zeros(p2,m2)    zeros(p2,p2)];
```

```
%con_rt = cond(rt)
ay = at - bt1*dd1'*(rt\ct);
cy = bt1*bt1' - bt1*dd1'*(rt\dd1)*bt1';
%if norm(cy)<sqrt(eps)
%   cy = zeros(n);
%end
%
% Solution of Y-Riccati equation
%
[y,l,g,rr] = dare(at',ct',bt1*bt1',rt,bt1*dd1');
%[y,w,rcond,ferr] = ricdsolv(1,ay',cy,ct'*(rt/ct));
% y = dricc(at',ct',bt1*bt1',rt,bt1*dd1');
%[y,l,g,rr] = dare(ay',ct',cy,rt);
%[rcon,ferr] = drcon(at',ct'*(rt\ct),bt1*bt1',y);
ycond = cndricd(ay',cy,ct'*(rt\ct),y);
%rcon
%eig_y = eig(y)
%
% Accuracy test
%
%resy = y-at*y*at'-bt1*bt1'+(at*y*ct'+bt1*dd1')*((rt+ct*y*ct')\(at*y*ct'+bt1*dd1')');
%norm_resy = norm(resy)/norm(y)
%
st1 = dt11*dt11' - gamma^2*eye(m2) + ct1*y*ct1';
st2 = dt11*dt21' + ct1*y*ct2';
st3 = dt21*dt21' + ct2*y*ct2';
%con_st3 = cond(st3)
st = [st1 st2;st2' st3];
del_st = st1-st2*(st3\st2');
%eig_del_st=eig(del_st)
mt1 = bt1*dt11' + at*y*ct1';
mt2 = bt1*dt21' + at*y*ct2';
mt = [mt1 mt2];
%
%eig_at = eig(at-mt*(st\ct))
%
ak = at - (b2/v12)*(ct1 - (st2/st3)*ct2) - (mt2/st3)*ct2;
bk =  (mt2/st3) - (b2/v12)*(st2/st3);
ck = -v12\(ct1 - (st2/st3)*ct2);
dk = -v12\(st2/st3);
```

```
%
% Take into account D22 =/= 0
%
f = eye(m2) + dk*d22;
%con_f = cond(f)
ck = f\ck;
dk = f\dk;
ak = ak - bk*d22*ck;
bk = bk - bk*d22*dk;
%
P = ltisys(a,b,c,d);
K = ltisys(ak,bk,ck,dk);
cls = slft(P,K,p2,m2);
[al,bl,cl,dl] = ltiss(cls);
```

```
%function [cond,Omega,Theta,Pi] = cndricd(A,C,D,X)
%CNDRICD Quantities related to the conditioning of the
%        discrete-time matrix algebraic Riccati equation
%
%             C - X + A'*X*inv(eye(n) + D*X)*A = 0.
%
%        The condition number of Riccati equation is given by
%
%        cond = norm([Theta*norm(A,'fro'), Omega*norm(C,'fro'),
%                     -Pi*norm(D,'fro')])/norm(X,'fro')
%
%        where Omega, Theta and Pi are defined by
%
%        Omega = inv(kron(Ac',Ac') - eye(n^2)),
%        Theta = Omega*(kron(eye(n),Ac'*X) + kron(Ac'*X,eye(n))*W)
%        Pi = Omega*kron(Ac'*X,Ac'*X), Ac = inv(eye(n) + D*X)*A
%
%        and W is the vec-permutation matrix.
%

%        RICCPACK, 28.02.2000
%

function [cond,Omega,Theta,Pi] = cndricd(A,C,D,X)
n = max(size(A));
nora = norm(A,'fro');
norc = norm(C,'fro');
nord = norm(D,'fro');

Ac = (eye(n) + D*X)\A;
M = kron(Ac',Ac') - eye(n*n);

Omega = inv(M);

W = 0*eye(n*n);
for i = 1:n,
    for j = 1:n,
    W(j+(i-1)*n,i+(j-1)*n) = 1.;
    end
end
```

```
Theta = M\(kron(eye(n),Ac'*X) + kron(Ac'*X,eye(n))*W);
Pi = M\kron(Ac'*X,Ac'*X);

D1 = norc*Omega;
D2 = nora*Theta;
D3 = nord*Pi;
norx = norm(X,'fro');
if norx > 0
   cond = norm([D1, D2, -D3]) / norx;
else
   cond = 0;
end
```