H_∞ controller design on the COMPl$_e$ib problems with the Robust Control Toolbox for Matlab

Didier Henrion[1,2]

Draft of October 3, 2005

Abstract

The version R14SP1 of function `hinfsyn` (full-order H_∞ optimal controller synthesis) of the Robust Control System Toolbox for Matlab is benchmarked on the extensive set of linear models of the COMPl$_e$ib database. The main conclusion is that sometimes the achieved numerical results are surprising, if not disappointing. This illustrates the difficulty of designing efficient and numerically reliable computer-aided control system design (CACSD) tools.

1 Introduction

The COMPl$_e$ib library is a freely available Matlab package [11] collecting a large number of continuous-time linear time-invariant (LTI) system matrices in state-space format

$$\begin{array}{c|c|c} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ \hline C_2 & D_{21} & 0 \end{array} \qquad (1)$$

where the block partitioning follows the by-now standard conventions in state-space linear system control [7]. A companion user's guide to the COMPl$_e$ib library is available that contains brief descriptions of the underlying physical problems and their modeling. The library is partitioned into several collections originating from distinct areas of engineering.

[1]LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France

[2]Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 16627 Prague, Czech Republic

These examples were extracted from references in the control literature spanning the last three decades. They are meant as benchmark problems for control analysis and design algorithms, and especially for static output feedback, H_2 and H_∞ optimal design.

Unfortunately there is no documentation available on the performance achievable in closed-loop for each problem instance, such as for example the maximum stability margin or minimum H_2 or H_∞ norm. In the case of static state feedback or full-order[3] dynamic controller feedback, solving these design problems amounts at solving a numerical linear algebra problem, or at worst a quasi-convex optimization problem over linear matrix inequalities. In particular, the Control System Toolbox [6] and the Robust Control Toolbox [1] for Matlab contain several routines for solving these problems in a user-friendly way.

The objective of this note is then to report our computational experience in solving full-order H_∞ optimal controller design problems on the COMPl$_e$ib benchmark examples, using the `hinfsyn` function of Release 14 Service Pack 1 (R14SP1) of the Robust Control Toolbox for Matlab. By reporting the best achieved closed-loop H_∞ norms we aim at complementing the COMPl$_e$ib library. In particular, we hope that this will provide further motivation for developers to test their algorithms (for full-order but also reduced-order controller design) and compare their results with publically available data.

2 Design algorithms

Starting from Matlab 7 (Release 14) the function `hinfsyn` of the Robust Control System toolbox merges the previously existing Matlab routines for full-order H_∞ optimal controller design.

In all our experiments we used the default input parameter tunings of function `hinfsyn`, namely

- no initial lower and upper bounds
- relative error tolerance of 1%

on the optimal H_∞ norm.

When the design algorithms return successfully a stabilizing controller, we then compute the actual H_∞ norm achieved in closed-loop. For this we use the function `norm` overloaded for LTI systems in the Control System Toolbox, an implementation of the bisection algorithm of [3] and [4]. We use the default tuning of 1% for the relative error tolerance in computing the H_∞ norm.

For our H_∞ design experiments we have used the following algorithms

[3] By full-order controller we mean a controller of the same order as the system to be controlled

- algebraic Riccati equation (ARE)
- regularized ARE (RARE)
- linear matrix inequality (LMI)

that we now briefly describe.

2.1 ARE

The ARE method is directly implemented in function `hinfsyn`. It is based on the two Riccati formulae [7] with loop-shifting [14], which are solved using specialized tools of numerical linear algebra and a bisection algorithm. It is available under some technical assumptions and restrictions on the input data.

2.2 RARE

The RARE method has been implemented by Alexandre Megretski [12] as an upper-layer function calling standard routines of the Control System Toolbox and then the ARE method. The method consists in pre-conditioning and perturbing the input data before calling `hinfsyn`:

1. the system matrix is balanced using function `balreal` of the Control System Toolbox. This is an implementation of the Gramian-based input/output balancing of state-space realizations, see [13] and [10]
2. the obtained system matrix is regularized to

$$\begin{array}{c|ccc|c} A & B_1 & \varepsilon I & 0 & B2 \\ \hline C_1 & D_{11} & 0 & 0 & D_{12} \\ \varepsilon I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \varepsilon I \\ \hline C_2 & D_{21} & 0 & \varepsilon I & 0 \end{array}$$

   where ε is a small positive absolute perturbation
3. the ARE method (see above) is applied on the regularized data.

2.3 LMI

The LMI method is also directly implemented in function `hinfsyn`. This convex optimization approach was developed partly to alleviate some of the restrictions of the ARE method. It was independently proposed in [8] and [9], see [16] and [15] for good overviews. As explained in [15] there are alternative LMI methods for full-order H_∞ design. However, up to our knowledge they are not implemented under Matlab.

3 Numerical results

We closely follow the problem classification used in COMPl$_e$ib . For each problem we indicate the open-loop system order, and the H_∞ norm achieved by full-order controllers designed via

- the ARE method with default tunings,
- the RARE method with regularization parameters $\varepsilon = 10^{-3}$ and 10^{-6},
- the LMI method with default tunings.

Since the relative tolerances for convergence of the H_∞ controller design algorithms and the H_∞ norm computation are set by default to 1% (see previous section), the achieved H_∞ performance is reported with 3 significant digits only.

In order to indicate that a method failed or returned a warning message, we use the identifiers described in Table 1.

name	type	diagnostic
modes	warning	controller has fast modes
unstable	failure	closed-loop is unstable
data	failure	assumptions on input data are not satisfied
eig	failure	cannot order eigenvalues
error	failure	other error messages
empty	failure	no error message but empty output
time	abort	execution time exceeds 5 minutes

Table 1: Identifiers for warning and error messages.

For systems of order greater than 20, we also report respective execution times for each method. We did not experiment with systems of order greater than 60.

Experiments are carried out with Matlab 7 (Release 14 Service Pack 1) on a PC computer with Pentium IV 1.6 Mhz processor with 512 Mb RAM.

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
AC1	5	**data**	$5.22 \cdot 10^{-7}$	$5.31 \cdot 10^{-4}$	$3.18 \cdot 10^{-6}$
AC2	5	**unstable**	0.112	0.112	0.111
AC3	5	2.98	2.97	2.98	2.96
AC4	4	0.562	0.562	0.563	0.558
AC5	4	658	**unstable**	658	658
AC7	9	0.0390	0.0459	0.0463	0.0380
AC8	9	1.62	1.62	1.62	1.62
AC9	10	1.01	1.00	1.00	1.00
AC10	55	**empty**	**eig**	3.07 (21s)	**time**
AC11	5	2.78	2.81	2.81	2.81
AC12	4	**unstable**	$2.20 \cdot 10^{-3}$	0.322	0.0215
AC13	28	156 (39s)	156 (8s)	156 (7s)	156 (162s)
AC14	40	100 (4s)	100 (9s)	100 (9s)	**time**
AC15	4	14.9	14.9	14.9	14.9
AC16	4	14.9	14.9	14.9	14.9
AC17	4	6.61	6.61	6.61	6.61
AC18	10	**unstable**	**empty**	7.92	5.39 (**modes**)

Table 2: Achieved H_∞ performance for aircraft models

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
HE1	4	0.0779	0.0783	0.0795	0.0738
HE2	4	**unstable**	2.42	2.42	2.42
HE3	8	**unstable**	0.806	0.806	0.800
HE4	8	22.9	22.8	22.8	22.8
HE5	8	1.77	1.77	1.77	1.77
HE6	20	2.39 (2s)	2.39 (2s)	2.39 (2s)	2.39 (27s)
HE7	20	2.61 (2s)	2.61 (2s)	2.61 (2s)	2.61 (28s)

Table 3: Achieved H_∞ performance for helicopter models

4 Discussion

Over the 111 examples, there are 2 examples for which all the methods fail, namely REA4 and LAH:

- REA4 is actually a discrete-time model, and should be converted to continuous-time or removed in future versions of COMPl$_\text{e}$ib . We do not consider this model in the discussion below;
- LAH is a numerically challenging 48th order SISO model of a building at the Los Angeles University Hospital, with flexible modes.

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
JE1	30	3.85 (42s)	3.89 (6s)	3.88 (6s)	3.86 (300s)
JE2	21	42.4 (29s)	53.7 (3s)	44.4 (3s)	42.6 (66s)
JE3	24	2.89 (2s)	2.89 (3s)	2.89 (3s)	2.89 (269s)
REA1	4	0.863	0.864	0.865	0.862
REA2	4	**unstable**	1.13	1.14	1.13
REA3	12	74.3	73.5	74.3	74.3
REA4	8	**error**	**error**	**error**	**error**

Table 4: Achieved H_∞ performance for jet engine (JE) and reactor (REA) models

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
DIS1	8	4.16	4.16	4.16	4.16
DIS2	3	**unstable**	0.952	0.954	0.949
DIS3	6	**unstable**	1.05	1.04	1.04
DIS4	6	**unstable**	0.733	0.734	0.732
DIS5	4	666	666	666	667
EB1	10	3.11	3.11	3.11	3.10
EB2	10	1.78	1.78	1.78	1.77
EB3	10	1.80	1.80	1.80	1.80
EB4	20	1.80 (2s)	1.80 (2s)	1.80 (2s)	1.80 (22s)
EB5	40	1.86 (8s)	1.80 (9s)	1.80 (8s)	**time**

Table 5: Achieved H_∞ performance for decentralized interconnected systems (DIS) and Euler-Bernoulli beams (EB)

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
TG1	10	3.47	3.47	3.47	3.47
AGS	12	8.17	8.19	8.17	8.17
WEC1	10	3.64	3.64	3.64	3.64
WEC2	10	3.60	3.60	3.60	3.60
WEC3	10	3.77	3.77	3.77	3.77
BDT1	11	0.271	0.270	0.270	0.266
MFP	4	4.20	4.19	4.19	4.20
UWV	8	$3.05 \cdot 10^{-8}$	$6.49 \cdot 10^{-7}$	$8.30 \cdot 10^{-4}$	**error**
IH	21	$5.19 \cdot 10^{-7}$ (15s)	**empty**	0.930 (5s)	$1.748 \cdot 10^{-2}$ (69s)
CSE1	20	0.0224 (5s)	0.0203 (1s)	0.0214 (1s)	0.0199 (49s)
CSE2	60	0.0226 (54s)	0.0209 (16s)	0.0269 (23s)	**time**
PAS	5	0.0104	1.08	119	0.00700
TF1	7	**data**	0.250	0.250	0.247
TF2	7	**data**	$5.20 \cdot 10^3$	$5.20 \cdot 10^3$	$5.20 \cdot 10^3$
TF3	7	**data**	0.250	0.371	0.247
PSM	7	0.920	0.920	0.920	0.920

Table 6: Achieved H_∞ performance for various physical systems

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
NN1	3	13.1	13.1	13.1	13.1
NN2	2	1.77	1.77	1.77	1.77
NN3	4	7.91	7.93	8.54	16.0 (**modes**)
NN4	4	**unstable**	1.29	1.29	1.29
NN5	7	238	238	238	238
NN6	9	128	127	127	127
NN7	9	33.0	33.1	33.1	33.1
NN8	3	**unstable**	2.36	2.36	2.36
NN9	5	**unstable**	14.5	13.6	13.7 (**modes**)
NN10	8	**unstable**	**unstable**	13.6	0.00
NN11	16	0.0163	0.0198	0.0589	0.0279
NN12	6	6.30	6.29	6.30	6.31
NN13	6	10.2	10.2	10.2	10.2
NN14	6	9.44	9.44	9.44	9.46
NN15	3	0.101	0.103	0.103	0.0978
NN16	8	**unstable**	1.22	0.956	0.956
NN17	3	2.64	2.65	2.64	2.64 (**modes**)

Table 7: Achieved H_∞ performance for academic test problems

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
HF2D10	5	$7.95 \cdot 10^4$	$7.95 \cdot 10^4$	$7.95 \cdot 10^4$	**error**
HF2D11	5	$7.64 \cdot 10^4$	**eig**	$7.64 \cdot 10^4$	**unstable**
HF2D12	5	$1.04 \cdot 10^6$	$1.04 \cdot 10^6$	$1.04 \cdot 10^6$	**unstable**
HF2D13	5	$1.02 \cdot 10^5$	$1.02 \cdot 10^5$	$1.02 \cdot 10^5$	$1.02 \cdot 10^5$ (**modes**)
HF2D14	5	$5.26 \cdot 10^5$	**eig**	$5.26 \cdot 10^5$	$5.97 \cdot 10^5$
HF2D15	5	$1.73 \cdot 10^5$	**eig**	$1.73 \cdot 10^5$	$2.32 \cdot 10^5$
HF2D16	5	$4.44 \cdot 10^5$	**eig**	$4.44 \cdot 10^5$	$6.45 \cdot 10^5$
HF2D17	5	$3.00 \cdot 10^5$	$3.00 \cdot 10^5$	$3.00 \cdot 10^5$	$3.00 \cdot 10^5$ (**modes**)
HF2D18	5	93.2	93.2	93.2	93.2 (**modes**)
HF2D_CD4	7	24.5	24.5	24.5	24.5
HF2D_CD5	7	25.8	25.8	25.8	25.8
HF2D_CD6	7	15.9	15.9	15.9	15.9 (**modes**)
HF2D_IS5	5	$1.13 \cdot 10^6$	$1.13 \cdot 10^6$	$1.13 \cdot 10^6$	**unstable**
HF2D_IS6	5	$1.32 \cdot 10^5$	$1.32 \cdot 10^5$	$1.32 \cdot 10^5$	**unstable**
HF2D_IS7	5	36.6	36.6	36.6	36.6
HF2D_IS8	5	91.3	91.3	91.3	91.3

Table 8: Achieved H_∞ performance for 2D heat flow models

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
CM1	20	0.820 (12s)	0.823 (2s)	0.823 (2s)	0.817 (29s)
CM2	60	0.820 (142s)	0.823 (21s)	0.823 (20s)	**time**
CM1_IS	20	0.820 (12s)	0.819 (2s)	0.818 (2s)	0.816 (42s)
CM2_IS	60	0.820 (148s)	0.822 (21s)	0.821 (19s)	**time**
TMD	6	**data**	2.12	2.12	2.12
FS	5	$7.79 \cdot 10^4$	$7.80 \cdot 10^4$	$7.79 \cdot 10^4$	$7.79 \cdot 10^4$
DLR1	10	0.0619	0.0619	0.0619	0.0619
DLR2	40	82.6 (106s)	82.6 (11s)	127 (15s)	**time**
DLR3	40	82.1 (81s)	82.6 (14s)	82.1 (12s)	**time**
LAH	48	**empty**	**empty**	**empty**	**time**

Table 9: Achieved H_∞ performance for second order models

problem	order	ARE	RARE 10^{-6}	RARE 10^{-3}	LMI
ROC1	9	**data**	1.13	1.13	1.13
ROC2	10	**data**	0.0410	0.0460	0.0413
ROC3	11	42.7	42.5	42.7	42.7
ROC4	9	**data**	8.97	23.2	16.1 (**modes**)
ROC5	7	**data**	$2.94 \cdot 10^{-6}$	$1.60 \cdot 10^{-3}$	$4.06 \cdot 10^{-5}$
ROC6	5	**data**	21.5	21.5	21.5
ROC7	5	**data**	1.12	1.12	1.12
ROC8	9	**data**	3.49	3.48	3.49
ROC9	6	**data**	2.24	2.26	2.24
ROC10	6	**data**	0.0799	0.0845	0.0754

Table 10: Achieved H_∞ performance for reduced order controller models

Generally speaking, we can observe that the less reliable method for H_∞ design is ARE. On the other hand, the RARE method with $\varepsilon = 10^{-3}$ fails only on the LAH model. Respective failure rates are, in decreasing order, $30/110 = 27\%$ for ARE, $15/110 = 14\%$ for LMI, $10/110 = 9\%$ for RARE ($\varepsilon = 10^{-6}$), and $1/110 = 1\%$ for RARE ($\varepsilon = 10^{-3}$). Note however that these rates must be interpreted carefully: the fact that a method did not fail does not necessarily imply that it returned a meaningful result.

In terms of computational burden, for systems of order greater than 20, the RARE method is generally significantly faster than the ARE method (see AC13, JE1, JE2, CSE2 and many of the second order models). Convergence of the bisection scheme is improved by regularizing the problem, except in a very few cases (see AC14, JE3). On the other hand, the LMI method is significantly more demanding. Compared with the other methods, the LMI method cannot deal with systems of order greater than 40 in a reasonable amount of time. This could be expected since the convex optimization formulation of the H_∞ design problem is much more general than the algebraic matrix formulation.

The tricky aspect of the RARE method is that sometimes there is no obvious choice for regularization parameter ε. Generally speaking, when the RARE method fails for $\varepsilon = 10^{-6}$, then it produces a controller for $\varepsilon = 10^{-3}$ (see AC5, AC10, AC18, IH, NN10, HF2D11, HF2D14, HF2D15, HF2D16). Yet sometimes the achieved H_∞ performance significantly varies as a function of ε, and it is not clear what is the actual value (see JE2, CSE1, NN3, NN9, NN11, ROC10 and mainly AC12, PAS, TF3, NN16, DLR2, ROC4 for which the obtained values are inconsistent).

Note also that the LMI method, even though apparently more reliable, also returns sometimes results which are inconsistent (see AC7, AC12, AC18, HE1, JE2, IH, PAS, NN3, NN10, NN11, HF2D14, HF2D15, HF2D16, ROC4, ROC10). It could be due to potential ill-conditioning of the problems. Note however that we do not feel comfortable with this notion, as we are not aware of any satisfying, computationally meaningful, and algorithmic independent measure of conditioning of an H_∞ design problem formulation. Interestingly, numerical troubles can be encountered already on low order systems (see AC12, PAS, ROC4).

Another aspect that must be taken into account is the fact that the H_∞ performance achieved by respective methods is evaluated numerically a posteriori on the closed-loop system with the bisection algorithm implemented in the `norm` function overloaded for LTI systems in the Control System Toolbox. We do not preclude numerical failures by this implementation, and so the values reported in the above table may be sometimes erroneous not because of the design method, but because of the norm computation algorithm.

5 Conclusion

In this note we described a series of H_∞ full-order controller design experiments carried out on benchmark problems of the COMPl$_e$ib collection. Our primary objective was to report the values of the H_∞ norm achieved by implementations of various design algorithms. We restricted ourselves to the methods implemented in the `hinfsyn` function of the new version of the Robust Control Toolbox for Matlab.

Our conclusions are that significant development efforts are still needed to improve the error and failure diagnostic mechanisms in the current implementations. For some examples, we have not been able to find consistent values of achievable H_∞ norms.

Alternative implementations of the ARE method for H_∞ full-order controller design, as well as the bisection algorithm for H_∞ norm computation, are available in the SLICOT library [2]. A comprehensive study and comparison with these tools could be an interesting research direction.

Developers of the Robust Control Toolbox for Matlab informed us in September 2005 that several of the issues described in this note have been fixed in the next release of function `hinfsyn`.

For the time being, we are planning to use the H_∞ norm values reported in this note as a comparative basis for our experiments on reduced-order H_∞ controller design using non-smooth optimization [5].

Acknowledgments

This work benefited from insightful discussions with Denis Arzelier, Bill Helton, Friedemann Leibfritz, Alexandre Megretski, Dimitri Peaucelle, Emmanuel Prempain, Carsten Scherer and Matthew Turner. Many thanks also to Michael Overton for his insightful experiments on nonsmooth optimization for fixed-order controller design.

References

[1] G. Balas, R. Chiang, A. Packard, M. Safonov. Robust Control Toolbox for use with Matlab. Version 3.0.1. The MathWorks, Inc. 2005.

[2] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, A. Varga. SLICOT - a subroutine library in systems and control theory. In B. N. Datta (Editor). Applied and computational control, signal and circuits. Vol. 1, Ch. 10, pp. 499-539, Birkhauser, Switzerland, 1999.

[3] S. Boyd, V. Balakrishnan, P. Katamba. A bisection method for computing the H_∞ norm of a transfer matrix and related problems. Mathematics of Control, Signals and Systems, Vol. 2, pp. 207–219, 1989.

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

[5] J. V. Burke, D. Henrion, A. S. Lewis, M. L. Overton. HIFOO - A Matlab package for fixed-order controller design and H_∞ optimization", to be registered as a LAAS-CNRS Research Report, Toulouse, France, September 2005. Submitted to the IFAC Symposium on Robust Control Design, Toulouse, July 2006.

[6] Control System Toolbox for use with Matlab. User's Guide. Version 6.2. The MathWorks Inc., USA, 2005.

[7] J. C. Doyle, K. Glover, P. Khargonekar, B. Francis. State-space solutions to standard H_2 and H_∞ control problems. IEEE Transactions on Automatic Control, Vol. 34, No. 8, pp. 831-847, 1989.

[8] P. Gahinet, P. Apkarian. A linear matrix inequality approach to H_∞ control. International Journal of Robust and Nonlinear Control, Vol. 4, No. 4, pp. 421-448, 1994.

[9] T. Iwasaki, R. E. Skelton. All controllers for the general H_∞ control problem: LMI existence conditions and state space formulas. Automatica, Vol. 30, No. 8, pp. 1307-1317, 1994.

[10] A. J. Laub, M. T. Heath, C.C. Paige, R.C. Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. IEEE Transactions on Automatic Control, Vol. 32, pp. 115-122, 1987.

[11] F. Leibfritz. $\mathrm{COMPl_eib}$: constraint matrix optimization problem library - a collection of test examples for nonlinear semidefinite programs, control system design and related problems. Technical Report, University ot Trier, Germany, 2004. See also `www.complib.de`

[12] A. Megretski, personal communication, 2005.

[13] B. Moore. Principal component analysis in linear systems: controllability, observability and model reduction. IEEE Transactions on Automatic Control, Vol. 26, pp. 17-31, 1981.

[14] M. G. Safonov, D. J. N. Limebeer, R. Y. Chiang. Simplifying the H_∞ via loop shifting, matrix pencil and descriptor concepts. International Journal of Control, Vol. 50, No. 6, pp. 2467-2488, 1989.

[15] C. W. Scherer, S. Weiland. Course on linear matrix inequalities in control. Delft University of Technology, The Netherlands, 2004.

[16] R. E. Skelton, T. Iwasaki, K. Grigoriadis. A unified algebraic approach to linear control design. Taylor and Francis, UK, 1998.