Fixed-Structure Discrete-Time $\mathcal{H}_\infty$ Controller Synthesis with HIFOO

Andrey P. Popov*, Herbert Werner* and Marc Millstone†‡

* Institute of Control Systems, Hamburg University of Technology, D-21073 Hamburg, Germany
{andrey.popov, h.werner}@tu-harburg.de
† Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA
‡ currently at IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
millstone@cims.nyu.edu

***Abstract*— This paper presents an extension of the HIFOO toolbox for Matlab for fixed-structure and fixed-order $\mathcal{H}_\infty$ controller design to discrete-time controller design. The approach is applied to a restricted complexity controller synthesis problem for an active suspension system.**

I. INTRODUCTION

The $\mathcal{H}_\infty$ control framework is well suited for designing optimal and robust controllers for linear time-invariant systems. If no constraints are imposed on the controller, the design problem is convex and can be efficiently solved (e.g. using linear matrix inequalities (LMIs)), but delivers controllers of the same order as the generalized plant. For control applications, where due to physical constraints simple (low-order or fixed-structure) controllers are needed, the design results in a non-convex problem with multiple local minima. Different direct optimization techniques and iterative LMI techniques have been used to attack this problem, with varying success and performance. Recently, in [1], [2] and [3] gradient-based approaches have been proposed for solving the problem in the continuous-time case, which although based on local search methods, are reported to deal well with the non-convex underlying problem. The former one will be used in the following, since it is readily implemented as a toolbox for Matlab - HIFOO [4].

The above mentioned restrictions on the controller structure/order are often accompanied by a limited sampling frequency, but demanding performance requirements. A continuous-time controller design for such systems might be misleading since potential problems like frequency warping [5] are not considered, or it might not be possible to impose desired restrictions on the controller, such as properness. This paper addresses the above problems by extending HIFOO to the design of fixed-structure discrete-time controllers, such that the whole design is carried out in discrete-time domain.

The paper is structured as follows. The modifications to the HIFOO algorithm to handle discrete-time controller synthesis are discussed in Section II. Section III illustrates the discrete-time design with an example, and Section IV concludes the paper.

The following notation is used: j is the complex unit; $\mathbb{R}^{n\times m}$ is the set of $n \times m$ real matrices; I is an identity matrix with an appropriate dimension; for a complex vector z, $\Re(z)$ denotes the real part and $z^\star$ the conjugate transpose (Hermitian); $\mathcal{F}_L(G(z), K(z))$ denotes the lower linear fractional transformation of system $G(z)$ with system $K(z)$ [6].

II. DISCRETE-TIME HIFOO

Consider the discrete-time (generalized) plant $G(z)$ for which a controller $K(z) \in \mathcal{K}(z)$ has to be designed, such that the $\mathcal{H}_\infty$ norm of the closed-loop system $\gamma = \|T(z)\|_\infty$ is minimized, where $\mathcal{K}(z)$ is the set of all controllers with the allowed structure and $T(z) = \mathcal{F}_L(G(z), K(z))$.

Since the $\mathcal{H}_\infty$ norm is defined only for stable systems, a controller should guarantee the system stability and minimize the norm. Note that this is implicitly taken care of when using LMIs, since the existence of a Lyapunov matrix satisfying the LMI conditions guarantees the closed-loop stability. This is however not the case when using gradient-based methods and hence one needs to split the design process into two steps: (1) search for a controller $K(s)$ stabilizing the closed-loop system and (2) minimization of the closed-loop $\mathcal{H}_\infty$ norm. Thus, for the first step one needs to exchange the gradient algorithm implemented in HIFOO for minimization of the spectral abscissa $\alpha(A_T)$ of the closed-loop system matrix A_T (maximal real part of the closed-loop eigenvalues) with a gradient algorithm for minimization of the spectral radius $\rho(A_T)$ (maximal absolute eigenvalue). Since the gradient computation of $\alpha(A_T)$ and the required chain rule applications leading to the dependence on $K(z)$ are documented elsewhere ([7], [8]), in the following it is only shown how to obtain the derivative of $\rho(A_T)$ with respect to (w.r.t.) the closed-loop system matrix A_T. From now on, the dependence on A_T is dropped for brevity.

Let $\lambda_i = \lambda_i(A_T)$ be the eigenvalues of the closed-loop system matrix $A_T \in \mathbb{R}^{n\times n}$, $i = 1, \ldots, n$. Let λ_k be the eigenvalue at which ρ is attained, i.e., $\rho = \max_i |\lambda_i| = |\lambda_k|$. Assume further that λ_k has algebraic multiplicity one. Thus to obtain the needed gradient of ρ w.r.t. the elements of A_T (and later w.r.t. $K(z)$) one has to find the gradient of $|\lambda_k|$ w.r.t. A_T. Because the modulus operation returns a real-valued gradient, whereas a complex one is needed here, the approach from [9] is used to represent $\lambda_k = \mathcal{R} + j\mathcal{I}$ as $\ell_k = \begin{bmatrix}\mathcal{R} & \mathcal{I}\end{bmatrix}$, that is as real two-dimensional vector. Note that

$$\rho = |\lambda_k| = \sqrt{\mathcal{R}^2 + \mathcal{I}^2} = |\ell_k|. \quad (1)$$

Thus, differentiating ρ as a function of $\mathcal{R}$ and $\mathcal{I}$ leads to

$$\rho' = \begin{bmatrix} \frac{\partial \rho}{\partial \mathcal{R}} & \frac{\partial \rho}{\partial \mathcal{I}} \end{bmatrix} = \begin{bmatrix} \frac{\mathcal{R}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}} & \frac{\mathcal{I}}{\sqrt{\mathcal{R}^2+\mathcal{I}^2}} \end{bmatrix} = \frac{\lambda_k}{|\lambda_k|}. \quad (2)$$

For a small number t and a matrix $A_T(t) = A_0 + Pt$ a well known result from eigenvalue perturbation theory [10], [11] states that if λ_k is an eigenvalue of $A_T(0)$ with corresponding right and left eigenvectors x and y, then

$$\frac{d\lambda_k}{dt} = \frac{y^\star \frac{dA_T(t)}{dt} x}{y^\star x}. \quad (3)$$

Using the chain rule one can combine (2) and (3) to

$$\frac{d\rho}{dt} = \Re \operatorname{trace}\left(\left(\frac{\lambda_k}{|\lambda_k|}\right)^\star \frac{d\lambda_k}{dt}\right) = \Re\left(\frac{\bar{\lambda}_k}{|\lambda_k|} \frac{y^\star \frac{dA_T(t)}{dt} x}{y^\star x}\right), \quad (4)$$

which also summarizes the necessary change in the first HIFOO step.

Since during the second step - the minimization of $\|T(z)\|_\infty$ - HIFOO relies on external computation of the $\mathcal{H}_\infty$ norm (e.g. the `norm` function in MATLAB) the only change needed is to account for the fact that the frequency range is $\omega \in (0, 2\pi)$ rather than $\omega \in (0, \infty)$. At the frequency ω_x at which the $\mathcal{H}_\infty$ norm is attained the closed-loop system gain will be

$$T_x = B_T(e^{jT_S\omega_x} I - A_T)^{-1} C_T + D_T,$$

where T_S is the sampling period. The computation of the gradient from this point on is identical to the continuous-time case ([7], [8]).

The discrete-time implementation of HIFOO, named HIFOOd, is based on HIFOO 2.0 and can be obtained from http://www.bitbucket.org/andrey.popov/hifoo-d. Both HIFOO and HIFOOd are distributed under the General Public License (GPL) version 3.

III. Numerical Example

For the purpose of illustration the Active Suspension System (ASS) benchmark problem [12] is considered. A schematic diagram of the system is shown in Fig. 1. The key

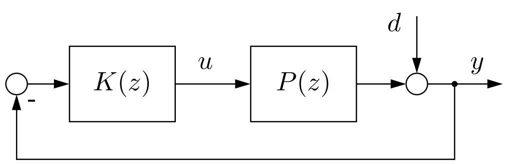

Fig. 1. Block diagram of the active suspension system.

idea of the ASS is as follows. During its operation a machine generates vibration that can be treated as a disturbance force d. A sensor mounted on the base of the machine measures the vibration y and a controller $K(z)$ is used to drive an active component (piston, motor, etc.) in such a way that the vibration is attenuated. Here $P(z)$ is called the secondary path [12] - from the active element to y. The model of the ASS and the design constraints are online available under http://lawww.epfl.ch/page11534.html. The design task is to design a discrete-time controller of as low order as possible that satisfies requirements on the sensitivity and control sensitivity functions of the system. A collection of different design approaches to this problem can be found in the November 2003 issue of the *European Journal of Control* [13].

For the purpose of comparison here the same generalized plant and weighting filters are used as in [14], [15], [16]. Because of a design requirement that the controller gain is equal to zero at frequency of $0.5F_s$, where $F_s = 800$ Hz is the sampling frequency, an additional term of $\frac{z+1}{z}$ is added to the system $P(z)$ during the design (to obtain $\hat{P}(z)$) and later is included in the controller. The closed-loop system with the generalized plant is shown in Fig. 2. The transfer functions of the shaping filters are given in the Appendix. The resulting generalized plant is of 27th order, which using standard $\mathcal{H}_\infty$ synthesis tools results in a 27th order controller.

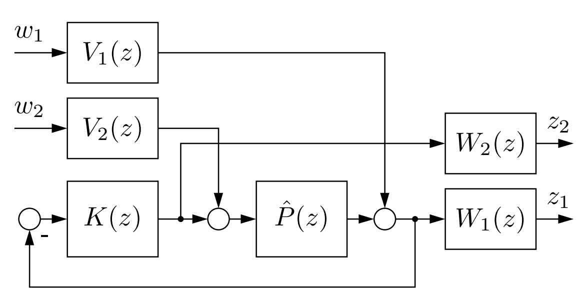

Fig. 2. Four-block generalized plant for mixed-sensitivity design.

In [14], [15] a full-order synthesis for the ASS is compared to a 5th order controller obtained by three other design techniques. In the first, a fixed-structure controller is obtained by applying a closed-loop weighted balanced reduction (BalRed) to the obtained full-order controller. The other two techniques rely, correspondingly, on curved line-search interior point (CLIP) method and a cone-complementarity (CC) method. A further approach, based on splitting the non-convex synthesis problem in a large-scale convex and a small-scale non-convex problems and solving them by a hybrid evolutionary-algebraic (HEA) method is discussed in [16].

Using the discrete-time HIFOO-algorithm presented in the previous section, 1st, 2nd and 5th order controllers were designed for the active suspension system. Furthermore the discrete-time generalized plant was transformed via bilinear (Tustin) transformation to continuous-time where the continuous time HIFOO-algorithm was used. Using a bilinear transformation the obtained controllers were transformed back to discrete-time and their performance was evaluated. Each of the designs was performed 5 times, each time with two randomly chosen initial points, and the mean values are reported.

The results are listed in Table I, where γ is the $\mathcal{H}_\infty$ norm of the closed-loop system in Fig. 2, k is the controller order, and where the results for the first 6 controllers are taken from [14], [15] and [16]. In order to compare the computation costs of the different designs, besides the computation times reported in [14], [15], [16], the full-order synthesis is repeated and a scaled value by its computation time – computation load – is provided.

TABLE I
PERFORMANCE AND COMPUTATION TIME OF CONTROLLERS

Controller design	k	γ	Computation time	Computation load
Full order [14]	27	2.476	11.9 s	1
BalRed [14]	5	3.405	12.8 s	1.07
CLIP [14]	5	2.506	4h23m45s	1329.00
CC [14]	5	2.630	14m33s	73.36
HEA [16]	5	2.589	2m42s	13.61
HEA [16]	2	2.596	1m25s	7.14
Full order	27	2.461	1.91 s	1
HIFOO - continuous	5	2.612	34.8 s	12.98
	2	2.582	18.1 s	9.47
	1	2.612	12.7 s	6.65
HIFOO - discrete	5	2.469	1m37s	50.78
	2	2.471	30.3 s	15.86
	1	2.612	19.2 s	10.05

Several conclusions can be drawn from Table I. Firstly, the discrete-time HIFOO synthesis achieves better results in terms of performance than the previously reported results and the continuous-time HIFOO synthesis. Secondly, the computational load with HIFOO (in particular the continuous-time synthesis) is lower than the one of the BalRed, CLIP and CC methods and comparable to the computational load with HEA. Finally, increasing the controller order generally leads to a better performance, but at the price of longer computation time – due to the increased problem size. In the case of the 5th order controller design using the bilinear transformation and the continuous-time HIFOO-algorithm the algorithm converged to a local minima with $\gamma \approx 2.612$ in all 5 runs. Only when initialized with the result from the 2nd order synthesis it managed to improve to $\gamma = 2.472$.

The sensitivity and control sensitivity plots of the full-order controller and the discrete-time HIFOO controllers providing the median closed-loop $\mathcal{H}_\infty$ norm among the 5 syntheses, are shown in Fig. 3 and Fig. 4 correspondingly.

The transfer functions of the 1st and 2nd order controllers designed by discrete-time HIFOO are

$$K_1(z) = \frac{0.02251z - 0.005877}{z - 0.9969}$$
$$K_2(z) = \frac{0.0335z^2 - 0.006038z + 0.01982}{z^2 - 0.5057z - 0.4893},$$

where, as discussed, an additional term $\frac{z+1}{z}$ has to be included.

Finally, in order to demonstrate the structural restrictions when a bilinear transformation is used in the design, consider

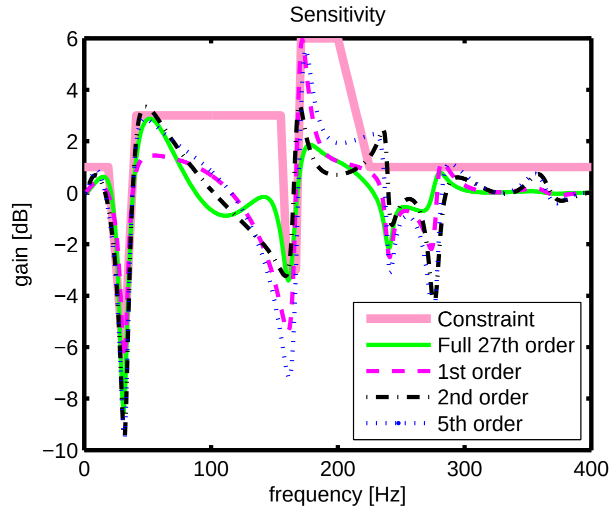

Fig. 3. Frequency response of the sensitivity $S(z) = (I + P(z)K(z))^{-1}$.

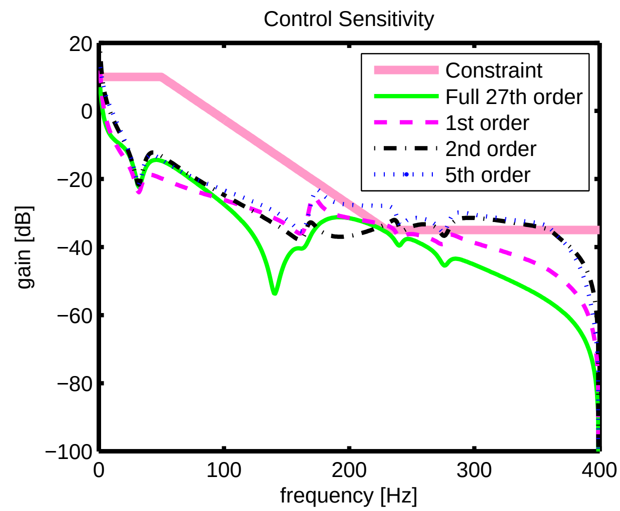

Fig. 4. Frequency response of the control sensitivity $\frac{u(z)}{d(z)} = -K(z)S(z)$.

that a first order strictly proper controller is desired, i.e. $D_K = 0$. The continuous time HIFOO algorithm returns the controller $K(s) = \frac{8.953}{s+38.19}$ which after discretization is transformed to the bi-proper controller $K(z) = \frac{0.005465(z+1)}{z-0.9534}$. On the other hand using the discrete-time HIFOO one directly obtains a controller with the desired structure $K(z) = \frac{0.006474}{z-0.9311}$.

IV. CONCLUSIONS

This paper presents an extension of the fixed-order $\mathcal{H}_\infty$ synthesis toolbox HIFOO to discrete-time controller synthesis. An active suspension benchmark problem was used to illustrate the superior performance of the algorithm compared to previously proposed low-order synthesis methods and the advantage of specifying the controller structure over the continuous-time design based on bilinear transformation.

$$V_1(z) = \frac{0.3933z^4 - 0.687z^3 + 0.5953z^2 - 0.383z + 0.1377}{z^4 - 2.498z^3 + 3.022z^2 - 2.336z + 0.8914} \qquad V_2(z) = 0.01; \qquad W_1(z) = 1;$$

$$W_2(z) = \frac{56.23z^6 - 27.88z^5 - 8.221z^4 - 14.89z^3 - 3.625z^2 - 0.7008z - 0.09963}{z^6 + 0.8845z^5 + 0.4842z^4 + 0.1615z^3 + 0.03581z^2 + 0.004889z + 0.0002892}$$

APPENDIX

The shaping filters are given at the top of the following page. Note that the ones reported in [15] are the bilinearly transformed continuous shaping filters, whereas the ones actually used (given here) are obtained by "sample-and-hold" transformation of the continuous ones.

ACKNOWLEDGMENT

The authors would like to thank Camile Hol for providing the shaping filters for the active suspension system needed for the comparison.

REFERENCES

[1] J. Burke, D. Henrion, A. Lewis, and M. Overton, "HIFOO - a Matlab package for fixed-order controller design and H_∞ optimization," in *IFAC Symposium on Robust Control Design*, Toulouse, France, 2006.

[2] ——, "Stabilization via nonsmooth, nonconvex optimization," *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1760 –1769, nov. 2006.

[3] P. Apkarian and D. Noll, "Nonsmooth H_∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.

[4] S. Gumussoy, M. Millstone, and M. Overton, "H_∞ strong stabilization via HIFOO, a package for fixed-order controller design," in *Conference on Decision and Control*, 2008.

[5] K. Aström and B. Wittenmark, *Computer-Controlled Systems: theory and design*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1997.

[6] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice-Hall, N.J., USA, 1996.

[7] J. Burke, A. Lewis, and M. Overton, "A nonsmooth, nonconvex optimization approach to robust stabilization by static output feedback and low-order controllers," in *Proc. IFAC Symposium on Robust Control Design*, Milan, Italy, 2003.

[8] M. Millstone, "HIFOO 1.5: Structured control of linear systems with a non-trivial feedthrough," Master's thesis, New York University, 2006.

[9] B. Han, M. Overton, and T. Yu, "Design of hermite subdivision schemes aided by spectral radius optimzation," *SIAM Journal on Scientific Computing*, vol. 25, no. 2, pp. 643–656, 2003.

[10] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge: Cambridge University press, 1985.

[11] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[12] I. Landau, A. Karimi, L. Miskovic, and H. Prochazka, "Control of an active suspension system as a benchmark for design and optimization of restricted complexity controllers," *European Journal of Control*, vol. 9, no. 1, pp. 3–12, 2003.

[13] I. Landau, A. Karimi, and H. Hjalmarson, Eds., *European Journal of Control*, vol. 9, no. 1, 2003.

[14] C. Hol, C. Scherer, E. van der Meché, and O. Bosgra, "A nonlinear SDP approach to fixed-order controller synthesis and comparison with two other methods applied to an active suspension system," *European Journal of Control*, vol. 9, no. 1, pp. 13–28, 2003.

[15] ——, "A trust-region interior point approach to fixed-order control applied to the benchmark system," in *Proc. International Workshop: Design and Optimisation of Restricted Complexity Controllers*, Grenoble, France, 2003, pp. 1–6.

[16] A. Farag and H. Werner, "Fixed-order control of active suspension: a hybrid evolutionary-algebraic approach," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005, pp. 171–176.