



HAL
open science

Multiobjective Robust Control with HIFOO 2.0

Suat Gumussoy, Didier Henrion, Marc Millstone, Michael Overton

► **To cite this version:**

Suat Gumussoy, Didier Henrion, Marc Millstone, Michael Overton. Multiobjective Robust Control with HIFOO 2.0. 2009. hal-00385733v1

HAL Id: hal-00385733

<https://hal.science/hal-00385733v1>

Preprint submitted on 20 May 2009 (v1), last revised 25 May 2009 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multiobjective Robust Control with HIFOO 2.0¹

Suat Gumussoy², Didier Henrion³, Marc Millstone⁴, Michael L. Overton⁵

Keywords: robust control; multiobjective control; optimization

Abstract

Multiobjective control design is known to be a difficult problem both in theory and practice. Our approach is to search for locally optimal solutions of a nonsmooth optimization problem that is built to incorporate minimization objectives and constraints for multiple plants. We report on the success of this approach using our public-domain MATLAB toolbox HIFOO 2.0, comparing our results with benchmarks in the literature.

1 Introduction

Multiobjective control aims at designing a feedback control law meeting potentially conflicting specifications defined on various input/output channels.

In the context of linear systems, a standard approach to multiobjective control is the Lyapunov shaping paradigm proposed in the mid 1990s by Scherer et al. (1997), as an outgrowth of the LMI (linear matrix inequality) formalism of Boyd et al. (1994). Within this scope, multiobjective controller design boils down to semidefinite programming (linear programming over the cone of positive semidefinite matrices) provided all the closed-loop specifications are certified simultaneously by a unique quadratic Lyapunov function. Moreover, the controller is retrieved a posteriori via tedious linear algebra, and its order is equal to the order of the open-loop plant plus the order of the weighting functions, which can be quite high in practice, in contradiction with simplicity of implementability requirements of embedded control laws. Another computational approach to multiobjective control exploits the parametrization of all stabilizing controllers described e.g. by Vidyasagar (1985). Linear programming can be used in this context to design controllers, see e.g. Boyd and Barratt (1991), but they are typically of very high order.

¹The research of D. Henrion was partly supported by project MSM6840770038 of the Ministry of Education of the Czech Republic. The work of M. Millstone and M.L. Overton was partly supported by the U.S. National Science Foundation under grant DMS-0714321; views expressed in the paper are those of the authors and not of the NSF. The work of M.L. Overton was also partly funded by Université Paul Sabatier, Toulouse, France.

²Katholieke Universiteit Leuven, Department of Computer Science, Belgium. suat.gumussoy@cs.kuleuven.be

³LAAS-CNRS, University of Toulouse, France, and Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic. henrion@laas.fr

⁴Courant Institute of Mathematical Sciences, New York University, USA. millstone@cims.nyu.edu

⁵Courant Institute of Mathematical Sciences, New York University, USA. overton@cims.nyu.edu

Following a decade of research efforts, these restrictions have been gradually relaxed (distinct Lyapunov functions for distinct performance channels, parameter-dependent Lyapunov functions, decoupling between Lyapunov and controller variables, lower-order controller design) at the price of an increased computational burden. The ROMULOC (robust multiobjective control) toolbox is a recent public-domain MATLAB implementation of these techniques, see Peaucelle and Arzelier (2006).

Particular cases of multiobjective robust control problems include strong stabilization (where a plant must be stabilized by a controller which is itself stable, see Vidyasagar (1985)) or simultaneous stabilization (where a single controller must be found that stabilizes several plants, see Blondel (1994)). Most of the algorithms or heuristics available to solve these problems also typically result in very high order controllers.

In this paper, we introduce the new release 2.0 of our freely available package HIFOO, which is aimed at removing the above mentioned limitations in the context of multiobjective controller design. First, the controller order is fixed at the outset, allowing for lower-order controller design. Second, no Lyapunov or lifting variables are introduced to deal with the conflicting specifications. The resulting optimization problem is formulated on the controller coefficients only, resulting in a typically small-dimensional nonsmooth nonconvex optimization problem that does not require the solution of any large convex subproblems, relieving the computational burden typical of Lyapunov LMI techniques. Because finding the global minimum of this optimization problem may be hard, we use an algorithm that searches only for local minima. While no guarantee can be made about the behaviour of this algorithm, in practice it is often possible to determine an acceptable controller quite efficiently.

See also Apkarian et al. (2008) for nonsmooth nonconvex optimization techniques applied to multiobjective robust control. As far as we know no software implementation of these techniques is publicly available at present.

2 Problem Formulation

The i th generalized plant $P^i = (A^i, B^i, C^i, D^i)$ describes the state-space equations

$$\begin{aligned}\dot{x}^i(t) &= A^i x^i(t) + B_1^i w^i(t) + B_2^i u^i(t), \\ z^i(t) &= C_1^i x^i(t) + D_{11}^i w^i(t) + D_{12}^i u^i(t), \\ y^i(t) &= C_2^i x^i(t) + D_{21}^i w^i(t) + D_{22}^i u^i(t),\end{aligned}$$

where $A^i \in \mathbb{R}^{n^i \times n^i}$, $D_{12}^i \in \mathbb{R}^{p_1^i \times m_2}$, $D_{21}^i \in \mathbb{R}^{p_2 \times m_1^i}$, with other matrices having compatible dimensions. The signals (z^i, w^i, y^i, u^i) respectively represent the regulated outputs, the exogenous inputs (including disturbance and commands), the measured (or sensor) inputs, and the control inputs. Let N be the number of plants.

The problem is to choose a *single controller*

$$K = (A_K, B_K, C_K, D_K)$$

with state-space equations

$$\begin{aligned}\dot{x}_K(t) &= A_K x_K(t) + B_K y(t), \\ u(t) &= C_K x_K(t) + D_K y(t),\end{aligned}$$

where $A_K \in \mathbb{R}^{n_K \times n_K}$, with B_K, C_K, D_K having dimensions that are compatible with A_K and the generalized plant matrices. The controller order n_K is *fixed*, so it can be specified by the designer. The \mathcal{H}_∞ norm of the i th closed loop system is the norm of the transfer function from input w^i to output z^i ; see Zhou et al. (1996) for details. The *complex stability radius* is a useful alternative measure when no w^i and z^i performance channels are specified and hence the \mathcal{H}_∞ norm is not defined: for a stable closed loop system, this is the inverse of the largest 2-norm perturbation to the closed loop system that can be tolerated while guaranteeing that the perturbed system remains stable. The *spectral abscissa* of a closed loop system is the largest of the real parts of its poles (eigenvalues).

Let $\beta_j, j = 1, \dots, N$ each be a real number or ∞ , and consider the following optimization problem:

$$\begin{aligned}\min_K \quad & \max_{j=1, \dots, N} \{g_j(K) : \beta_j = \infty\} & (1) \\ \text{subject to} \quad & g_j(K) \leq \beta_j, \quad j = 1, 2, \dots, N, & (2)\end{aligned}$$

where each g_j is one of the following supported functions of the closed-loop system for P^j , abbreviated by a single letter as follows:

- 'h': \mathcal{H}_∞ norm (∞ if unstable);
- 'r': reciprocal of complex stability radius (∞ if unstable);
- 's': spectral abscissa.

The functions g_j are all nonconvex, nonsmooth functions of the controller matrices. Thus, the optimization problem is potentially quite difficult. Consider some scenarios, where ϵ is a fixed small positive number.

Scenario 1. All $\beta_j = -\epsilon$, and all g_j are 's'. Thus the problem is to stabilize all plants simultaneously, with the real parts of all poles being at most $-\epsilon$.

Scenario 2. All $\beta_j = \infty$, and all g_j are 's'. Thus the problem is to minimize the maximum of the real parts of the closed loop poles of all plants. This is another approach to simultaneous stabilization, since if the final objective value is negative, all closed loop plants are stable. This approach is perhaps more appealing than Scenario 1, since the goal is to move all poles as far left as possible in the minmax sense, making the closed loop plants “more stable.”

Scenario 3. All $\beta_j = \infty$, and all g_j are 'r'. The problem is to stabilize all plants and *maximize* the maximum of the complex stability radii of the closed loop plants. This is a third approach to simultaneous stabilization, since if the final value is finite, all closed loop plants are stable. This approach is perhaps more appealing than either Scenario 1

or 2, since the complex stability radius is by definition a more robust measure of stability than the distance from the poles to the imaginary axis, which is the focus of Scenario 2.

Scenario 4. All $\beta_j = \infty$, and all g_j are 'h'. The problem is to stabilize all plants and minimize the maximum of the \mathcal{H}_∞ norms of the closed loop plants. This fourth approach to simultaneous stabilization may be the most appealing of all, since the \mathcal{H}_∞ norm is generally considered a more practically important measure than either the complex stability radius or the spectral abscissa.

Scenario 5. $\beta_1 = \infty$, $\beta_2 = 1/\epsilon$, g_1 is 'h' and g_2 is 'r'. The problem is to stabilize both plants and minimize the \mathcal{H}_∞ norm of the first closed loop plant subject to the complex stability radius of the second being at least ϵ .

If one wishes to impose restrictions, such as stability, on the *controller* (so-called *strong stabilization*), it is possible to do so using the multiple-plant model above by defining a plant P^i so that the closed loop plant *is equivalent* to the system described by the controller. However, we provide a more convenient way to specify controller stability directly, as described below. Benchmarks assessing the value of a previous version of HIFOO for strong stabilization appear in Gumussoy et al. (2008).

3 Optimization method

HIFOO 2.0 searches for local minimizers of (1)-(2). The algorithm has two phases. In each phase the main workhorse is the BFGS optimization algorithm, which is surprisingly effective for nonconvex, nonsmooth optimization, see Lewis and Overton (2009). The user can provide an initial guess for the desired controller (see below); if this is not provided, HIFOO generates randomly generated initial controllers, and even when an initial guess is provided, HIFOO generates some additional randomly generated initial controllers in case they provide better results.

The first phase is *stabilization*: BFGS is used to minimize the maximum of the spectral abscissae of the closed loop plants for which g_j is either 'h' or 'r'. This process terminates as soon as a controller is found that stabilizes these plants, thus providing a starting point for which the objective function for the second phase is finite.

The second phase is *optimization*: BFGS is used to look for a local minimizer of the following unconstrained problem:

$$\min_K F(K) + \rho \sum_{j=1}^N \max(0, g_j(K) - \beta_j),$$

where F is the objective function defined in (1) and ρ is a positive penalty parameter multiplying the sum of the constraint violations. If BFGS is unable to find a point for which the constraint violations are zero, the penalty parameter ρ is increased and the optimization is repeated as needed (unless F is identically zero, that is all β_j are finite). Although there are no guarantees, very often this process is quite effective and reasonably

fast. HIFOO also provides an option to invoke the gradient sampling method of Burke et al. (2005) after BFGS terminates, but this is generally more time consuming.

4 User interface

HIFOO is written in MATLAB. The simple call

$$K = \text{hifoo}(P, \text{order})$$

looks for a controller K solving Scenario 4 above: stabilize the plants described by P and minimize the sum of the \mathcal{H}_∞ norms of the closed loop plants, using a controller of the specified order. Here P is a cell array of plants specified in any of several formats, typically using the `ss` class of the MATLAB Control Systems Toolbox. A useful abbreviation is `P{j}='K'`, which specifies that the closed loop system for the j th plant is the controller itself. If the `order` argument is omitted, the default order 0 is used (static output feedback).

A more general calling sequence is

$$[K, F, \text{viol}] = \text{hifoo}(P, \text{order}, \text{init}, \text{fun}, \dots \\ \text{upperbnd}, \text{options})$$

where P and `order` are as above, `init` is an initial guess for the controller (several formats are supported), `fun` is a string specifying the characters defining the supported functions g_j (see Section 2) or a single character if all g_j are the same (default: 'h'), `upperbnd` is an array specifying the upper bounds β_j (default: all bounds set to ∞), and `options` is a structure with various optional fields, some of which are described below. The order of the input arguments does not matter except that P must be first. The outputs are, in addition to the controller K , the value of the objective function $F(K)$ in (1) and a vector of constraint violations $\max(0, g_j(K) - \beta_j)$. There is a fourth output argument `loc` (“local optimality certificate”) that can be requested if gradient sampling is used in addition to BFGS.

Some of the more useful fields in `options` are:

- `options.cpumax`: requests HIFOO to quit when the CPU time in seconds exceeds this quantity (default: ∞)
- `options.fast`: 1 to use a fast optimization method only (BFGS), 0 (default) to finish optimization with slower method (gradient sampling, which may give a better answer)
- `options.prtlevel`: one of 0 (no printing), 1 (minimal printing, default), 2 or 3 (more verbose)

- `options.struct`: specifies *sparsity structure* to be imposed on the controller (see documentation for details)
- `options.weightNormK`: weight for adding a penalty on the size of the controller to the objective function, specifically $(\|A_K\|^2 + \|B_K\|^2 + \|C_K\|^2 + \|D_K\|^2)^{1/2}$ (default: 0)
- `options.augmentHinf`: weight for adding reciprocal of complex stability radius to \mathcal{H}_∞ norm function to avoid closed loop plants that are only marginally stabilized: applies to all plants for which g_j is 'h' (default 0)

HIFOO 2.0 uses the following external software:

- required: HANSO 1.0, a hybrid algorithm for nonsmooth optimization, freely available from the HIFOO web page;
- required: the MATLAB Control Systems Toolbox, for \mathcal{H}_∞ norm and complex stability radius computation; this also provides user-friendly system modeling with the `ss` class;
- optional: `quadprog` from either the MATLAB Optimization Toolbox or MOSEK; required only when gradient sampling is used in addition to BFGS

The `linorm` code that was used by HIFOO 1.0 as an alternative to the \mathcal{H}_∞ norm code in the MATLAB Control Systems Toolbox is unfortunately not currently available for the latest versions of MATLAB.

5 Benchmarks

Simultaneous stabilization is one of the challenging control problems in the literature. The objective is to stabilize multiple plants by a single controller. This is Scenario 1 with ϵ set to an arbitrarily small number. In practice, whether we set ϵ to a small number such as 10^{-16} or simply set ϵ to 0 makes no difference. However, it is generally preferable to use one of Scenarios 2, 3 or 4, to optimize a measure of stability or performance in addition to stabilization. In the benchmarks that follow, we use Scenario 2 (optimizing the spectral abscissa). Notice that we do not need to explicitly set the upper bounds to ∞ , since that is the default value.

The following MATLAB script illustrates the process. There are three first-order plants, given in Jia and Ackermann (2001), and we wish to stabilize them, using Scenario 2, with a first-order controller:

```
>> P = {ss(tf([2 -9],[1 -8.8])), ...
        ss(tf([1 2],[1 -6])), ...
        ss(tf([2.5 6],[1 -8]))};
```

```

>> K = hifoo(P,'s',1);
...
hifoo: best order 1 controller found
has spectral abscissa -0.284216
...
>> tf(K)
  Transfer function:
s + 1.182
-----
s + 1.595
>> eig(feedback(P{1},-K))
ans =
  -0.2842 + 1.8223i
  -0.2842 - 1.8223i
>> eig(feedback(P{2},-K))
ans =
  1.0e+009 *
  -4.5265
  -0.0000
>> eig(feedback(P{3},-K))
ans =
  -9.7628
  -1.4768

```

Note that a third-order controller was designed in Jia and Ackermann (2001).

The output of HIFOO may differ on different runs since the initialization is done randomly. However, the output from one run may be used to initialize a second run on the same problem; the result cannot be worse. For example, following the run above with

```

>> K = hifoo(P,K,'s',1);

```

results in

```

...
hifoo: best order 1 controller found
has spectral abscissa -0.286524

```

which is a slight improvement. Depending on the problem and the initial randomization, several successive calls to HIFOO may be required to obtain a stabilizing controller.

To evaluate HIFOO, we consider 31 benchmark problems for simultaneous stabilization as shown in Table 1. There are 11 problems from applications and 20 academic test problems.

Some of the problems can be grouped as specialized simultaneous stabilization problems,

- *Simultaneous Stabilization by State Feedback Controller*: F4E Aircraft, CRJ–200 Aircraft, Passive Fault Tolerant Control, Ship-Steering
- *Simultaneous Stabilization by Static Output Feedback Controller*: Gas Turbine Engine, Stirred-Tank Reactor, Oblique Wing Aircraft, Bhattacharyya et al Ex. 1-1, Cao-Sun Ex., Bredemann Ex. 5.6 and 5.7, Chen *et al.*, Ex., F.-Anaya *et al.*, Ex. 1 and Ex. 4.

Problem Name	$N \times (n^{\max}, m_2, p_2)$	Known	HIFOO
Gas Turbine Engine	$2 \times (10, 2, 5)$	0	0
F4E Fighter Aircraft	$4 \times (3, 1, 3)$	0	0
CRJ–200 Aircraft	$6 \times (6, 1, 6)$	0	0
PFTC	$6 \times (4, 2, 4)$	0	0
Ship-Steering	$2 \times (3, 1, 3)$	0	0
Stirred-Tank Reactor	$3 \times (2, 1, 1)$	0	0
Lane-Keeping of AV	$3 \times (2, 1, 1)$	4	1
Helicopter Toy	$4 \times (3, 1, 1)$	2	1
Servomotor	$4 \times (2, 1, 1)$	1	1
Oblique Wing Aircraft	$64 \times (4, 1, 1)$	0	0
RHM14	$4 \times (8, 4, 4)$	26	0
Bhattacharyya <i>et al.</i> , Ex. 1-1	$16 \times (3, 1, 1)$	0	0
Bhattacharyya <i>et al.</i> , Ex. 1-2	$16 \times (3, 1, 1)$	1	2
Bhattacharyya <i>et al.</i> , Ex. 2-1	$8 \times (2, 1, 1)$	0	1
Bhattacharyya <i>et al.</i> , Ex. 2-2	$8 \times (2, 1, 1)$	1	1
Cao-Sun	$3 \times (2, 1, 1)$	0	0
Jia-Ackermann	$3 \times (1, 1, 1)$	3	1
Bredemann, Ex 4.2	$3 \times (3, 1, 1)$	2	2
Bredemann, Ex 5.5	$3 \times (2, 1, 1)$	3	1
Bredemann, Ex 5.6	$3 \times (2, 1, 1)$	5	0
Bredemann, Ex 5.7	$3 \times (1, 1, 1)$	1	0
Chen <i>et al.</i>	$3 \times (2, 3, 1)$	6	0
Arehart-Wolovich	$3 \times (2, 1, 1)$	1	1
F.-Anaya <i>et al.</i> , Ex.1	$80 \times (3, 1, 1)$	4	0
F.-Anaya <i>et al.</i> , Ex.2	$3 \times (3, 1, 1)$	6	1
F.-Anaya <i>et al.</i> , Ex.3	$4 \times (4, 1, 1)$	3	1
F.-Anaya <i>et al.</i> , Ex.4	$5 \times (3, 1, 1)$	3	0
Günder-Kabuli	$5 \times (10, 2, 2)$	4	1
Henrion <i>et al.</i> , 1 st Ex.	$3 \times (1, 1, 1)$	1	0
Henrion <i>et al.</i> , 2 nd Ex.	$3 \times (1, 1, 1)$	1	0
Blondel <i>et al.</i>	$4 \times (1, 1, 1)$	1	0

Table 1: Benchmarks on simultaneous stabilization. The third and fourth columns show the lowest stabilizing controller orders in the literature and the lowest found by HIFOO.

Our benchmark results are given in Table 1, showing the problem name, dimensions (with $n^{\max} = \max\{n^i\}$), the lowest known controller order from the literature (third column),

and the controller orders obtained by HIFOO (final column). Sometimes multiple runs were used to obtain these results. We set `options.fast = 0` (the default) but this is not essential to obtain similar results.

The performance of HIFOO is very good considering the large variety and number of benchmark examples. HIFOO successfully stabilizes the simultaneous stabilization problems with a *low-order* single controller compared to the existing methods in the literature. Note that contrary to the other methods in the literature, HIFOO allows the user to set the controller order *a priori*.

In particular, HIFOO is very successful in application problems. The results are similar to the existing methods in the literature for Gas Turbine Engine, F4E Fighter Aircraft, CRJ–200 Aircraft, Passive Fault-Tolerant Control, Ship-Steering, Stirred-Tank Reactor, Oblique Wing Aircraft, Servomotor. HIFOO performs better than existing methods for the application benchmarks Lane-Keeping of Automated Vehicles and Helicopter Toy. The Rationalized Helicopter Model (RHM14) shows the conservativeness of some of the methods in the literature, which produce a 26th order stabilizing controller. In contrast, HIFOO stabilizes the same benchmark by a static controller.

HIFOO shows similar performance for academic test problems. For almost all problems, the results are better than or equivalent to the results of existing methods. Some of these are Bredemann Ex.5.6, Chen *et al.*, F.-Anaya *et al.* Ex.1 and Ex.2, and Gündes-Kabuli, for all of which the existing methods stabilize the benchmark problem with a high-order controller and HIFOO solves the same problem by a static or first-order controller.

There are two benchmark examples for which HIFOO performs slightly worse than the existing methods, namely Ex.1-1 and 2-1 from Bhattacharyya *et al.* which are interval plants. HIFOO stabilizes both benchmark problems with a controller whose order is one more than the order of the controllers in the literature.

These results clearly demonstrate that HIFOO is very effective over various types of benchmark examples including industrial application and academic test problems. In conclusion, HIFOO, by allowing the designer to specify the controller order, is a useful tool for simultaneous stabilization.

6 Webpage

Instructions to download and install HIFOO 2.0 can be found at

www.cs.nyu.edu/overton/software/hifoo

References

Apkarian, P., D. Noll and A. Rondepierre (2008). Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control via nonsmooth optimization. *SIAM J. Control Optim.*, 47(3):1516-1546.

- Blondel, V. D. (1994) *Simultaneous stabilization of linear systems*. Springer.
- Boyd, S. P. and G. H. Barratt (1991). *Linear control design: limits of performance*. Prentice Hall.
- Boyd, S. P., L. El Ghaoui, E. Feron and V. Balakrishnan (1994). *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- Burke, J. V., A. S. Lewis and M. L. Overton (2005). A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM J. Optimization*, 15:751-779.
- Burke, J. V., D. Henrion, A. S. Lewis and M. L. Overton (2006). HIFOO - A Matlab package for fixed-order controller design and H-infinity optimization. *Proc. IFAC Symp. Robust Control Design*, Toulouse, France.
- Lewis, A. S. and M. L. Overton (2009). Nonsmooth optimization via BFGS. *Submitted to SIAM J. Optimization*.
- Doyle J., B. Francis and A. Tannenbaum (1990). *Feedback Control Theory*. Macmillan.
- Gumussoy, S., M. Millstone and M. L. Overton (2008). H-infinity strong stabilization via HIFOO, a package for fixed-order controller design. *Proc. IEEE Conf. Decision and Control*, Cancún, Mexico.
- Jia, Y. and J. Ackermann (2001). Condition and algorithm for simultaneous stabilization of linear plants. *Automatica* 37(9):1425-1434.
- Peaucelle, D. and D. Arzelier (2006). Robust multiobjective control toolbox. *Proc. IEEE Symp. Computer-Aided Control System Design*, Munich, Germany.
www.laas.fr/OLOCEP/romuloc
- Scherer, C. W., P. Gahinet and M. Chilali (1997). Multiobjective output feedback control via LMI optimization. *IEEE Trans. Autom. Control*, 42(7):896-911.
- Vidyasagar M. (1985). *Control system synthesis: a factorization approach*. MIT Press.
- Zhou, K., J. C. Doyle and K. Glover (1996). *Robust and optimal control*. Prentice Hall.