

Fixed-order output-feedback control design for LTI systems: a new algorithm to reduce conservatism

E. Simon^{a*}, P. Rodriguez-Ayerbe^b, C. Stoica^b, D. Dumur^b and V. Wertz^a

Abstract—This work proposes an algorithm to reduce the conservatism of fixed-order output-feedback control design for Linear Time Invariant (LTI) systems with Linear Matrix Inequalities (LMIs)-representable objectives. Using Lyapunov theory and the Schur complement many objectives can be written as Bilinear Matrix Inequalities (BMIs), which in general are hard to solve and have a non-convex space of solutions. The classical response to this is to use LMIs reformulation of BMIs, therefore using convex subspaces of the non-convex space of all solutions and thus introducing conservatism in general. Here a new use of a change of variables is proposed, so that consecutive convex subspaces are considered iteratively. This algorithm explores further the non-convex space of solutions, leading to improved objectives with reduced conservatism.

Index Terms—LMI, Control design, Output-feedback, LTI system

I. INTRODUCTION

The control design is made using the classical $P - K$ representation. P is the plant description, including the system model but also the observer structure and any filters or structures that are necessary to form the performance channels $T_{\omega_i z_j}$ and the control channel y_P, u_P . K is the control design parameter: $u_P = Ky_P$. All this is drawn with the following Fig. 1,

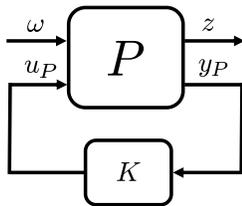


Fig. 1. Classical $P - K$ representation

with ω the input(s) of the objective(s) channel(s), z the output(s), y_P and u_P the input and output of the control parameter K .

P has a state-space representation given by:

$$\begin{cases} \dot{x} = Ax + B_i \omega_i + B u_P \\ z_j = C_j x + D_{ij} \omega_i + E_j u_P \\ y_P = Cx + F_i \omega_i + 0 \end{cases}$$

^aDepartment of Mathematical Engineering, Université Catholique de Louvain, 4 avenue Georges Lematre, 1348 Louvain-la-Neuve, Belgium Tel. +32 10478032 Fax +32 10472180 *Corresponding author. Email emile.simon@uclouvain.be; vincent.wertz@uclouvain.be.

^bSUPELEC Systems Sciences (E3S) Automatic Control Department, 3 rue Joliot Curie, 91192 Gif-sur-Yvette cedex, France, Tel. +33 169851375 Fax +33 169851389 {pedro.rodriguez; cristina.stoica; didier.dumur}@supelec.fr.

where x are the states of P . Note that \dot{x} is replaced by $x(k+1)$ in the discrete time case. The indexes i allow to designate each component of ω and the indexes j each component of z . The state-space representation of K is:

$$\begin{cases} \dot{x}_K = A_K x_K + B_K y_P \\ u_P = C_K x_K + D_K y_P \end{cases}$$

Closing the $P - K$ loop, the state-space representation of the closed-loop system $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ is obtained where each channel $T_{i_j} : \omega_i \rightarrow z_j$ has its realization:

$$\left(\begin{array}{cc|c} \mathbf{A} & \mathbf{B}_i & \\ \mathbf{C}_j & \mathbf{D}_{ij} & \end{array} \right) = \left(\begin{array}{cc|c} A + BD_K C & BC_K & B_i + BD_K F_i \\ BK_C & AK & BK_F i \\ \hline C_j + E_j D_K C & E_j C_K & D_{ij} + E_j D_K F_i \end{array} \right)$$

The state-space matrices of P must have the following structure, necessary for the change of variables reminded later:

$$\begin{aligned} A &= \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, & B_i &= \begin{bmatrix} B_{\omega_i \chi} \\ B_{\omega_i \epsilon} \end{bmatrix}, & B &= \begin{bmatrix} B_{u_P \chi} \\ 0 \end{bmatrix}, \\ C_j &= [C_{\chi z_j} \quad C_{\epsilon z_j}], & D_{ij} &= D_{\omega_i z_j}, & E_j &= D_{u_P z_j}, \\ C &= [0 \quad C_{\epsilon y_P}], & F_i &= D_{\omega_i y_P} \end{aligned} \quad (1)$$

The previous notations are close to that of e.g. [1]. The states of P are split here between: $-\epsilon$, the difference between the actual plant and observer states $-\chi$, the rest of the states. This representation is obtained using an observer (e.g. simply the plant model if stable) in P . The input y_P of the design parameter K has to be the difference between the actual plant output y and the observer output \hat{y} .

The difficulty will arise because of products between the Lyapunov matrix(ces) and the state-space representation of the design parameter. To illustrate this fact, we remind for example the LMI formulation of the \mathcal{H}_2 norm:

$$\|T(s)_{ij}\|_2 < \gamma_{ij} \text{ iff } \text{tr}(\mathbf{Z}_{ij}) < \gamma_{ij} \text{ and}$$

$$\left(\begin{array}{cc} \mathbf{A}' \mathbf{X}_{ij} + \mathbf{X}_{ij} \mathbf{A} & \mathbf{X}_{ij} \mathbf{B}_i \\ * & -\gamma_{ij} I \end{array} \right) \prec 0, \left(\begin{array}{cc} \mathbf{X}_{ij} & \mathbf{C}'_j \\ * & \mathbf{Z}_{ij} \end{array} \right) \succ 0, \mathbf{D}_{ij} = 0$$

The optimization variables are (within bold-faced characters): -the Lyapunov matrices $\mathbf{X}_{ij} = \mathbf{X}'_{ij} \succ 0$ associated to their channel T_{ij} , -the state-space matrices A_K, B_K, C_K, D_K , -the objectives variables γ_{ij} positive scalar, \mathbf{Z}_{ij} symmetric positive definite matrix with $\text{tr}(\mathbf{Z}_{ij})$ the trace of \mathbf{Z}_{ij} . The $'$ indicates the transposed matrix and the $*$ indicates a symmetrical term (all the BMIs and LMIs are symmetrical). One can easily observe the bilinear terms within the matrices. Therefore one has to use a change of variables in order to turn these troublesome BMIs into LMIs.

The change of variables of interest for this work is that of [1], [2], reminded now. Note that this is the second change of variables known of the two available in the literature, the first is that of [3], [4] (more on this later).

The change of variables

The BMIs have to be rewritten as follows: $\mathbf{X} \rightarrow \mathbf{X}(v)$, $\mathbf{XA} \rightarrow \mathbf{A}(v)$, $\mathbf{XB}_i \rightarrow \mathbf{B}_i(v)$, $\mathbf{C}_j \rightarrow \mathbf{C}_j(v)$ where each of these terms is defined here. $\mathbf{D}_{ij} = D_{ij} + E_j \mathbf{D}_K F_i$ does not change.

The Lyapunov matrices are changed (and restructured) into $\mathbf{X}_{ij} \rightarrow (R_{ij}, S_{1ij}, S_{2ij}, T_{11ij}, T_{12ij}, T_{22ij}) = (v)$ (more details in [1]). In order to lighten the notations, the ij indexes are omitted in all the following terms:

$$\begin{aligned} \mathbf{A}(v) &= \begin{pmatrix} A_1 R & t_1 & t_2 \\ 0 & T_{11} A_2 + T_{12} B_K C_{\epsilon y_P} & T_{12} A_K \\ 0 & T_{12}' A_2 + T_{22} B_K C_{\epsilon y_P} & T_{22} A_K \end{pmatrix} \\ t_1 &= A_1 S_1 - S_1 A_2 - S_2 B_K C_{\epsilon y_P} + A_3 + B_{u_P \chi} D_K C_{\epsilon y_P} \\ t_2 &= A_1 S_2 - S_2 A_K + B_{u_P \chi} C_K \\ \mathbf{B}(v) &= \begin{pmatrix} B_{u_P} D_K D_{\omega y_P} + B_{\omega \chi} - S_1 B_{\omega \epsilon} - S_2 B_K D_{\omega y} \\ T_{11} B_{\omega \epsilon} + T_{12} B_K D_{\omega y} \\ T_{12}' B_{\omega \epsilon} + T_{22} B_K D_{\omega y} \end{pmatrix}, \\ \mathbf{C}(v)' &= \begin{pmatrix} RC'_{xz} \\ S_1' C'_{xz} + C'_{\epsilon z} - C'_{\epsilon y_P} D_K' D'_{uz} \\ S_2' C'_{xz} - C_K' D'_{uz} \end{pmatrix}, \\ \mathbf{X}(v) &= \begin{pmatrix} R & 0 & 0 \\ * & T_{11} & T_{12} \\ * & * & T_{22} \end{pmatrix}, \end{aligned} \quad \Gamma_\beta \leq \Gamma_\alpha \quad (2)$$

So far this change of variables has only been used fixing the A_K, B_K matrices beforehand (using a polynomial expansion, like FIR filters in [1]). Indeed, this *a priori* choice is necessary since A_K, B_K appear non affinely with other variables. Considering this, a key observation can be made: only the variables S_2, T_{12}, T_{22} , related to the Lyapunov matrices, multiply the A_K, B_K matrices. This is at the basis of the contribution of this work, described in the next section.

II. CONTRIBUTION

The normal set of variables to be used with the change of variables of [1] is the following: $v_\alpha = C_K, D_K, R, S_1, T_{11}, S_2, T_{12}, T_{22}$. With this, one can find the best C_K, D_K (without conservatism) matrices for given A_K, B_K matrices (defined beforehand and therefore conservative in general). Instead of choosing A_K, B_K with a polynomial expansion as in [1] (without convenient equivalent in continuous-time), an alternate initial choice is made here. It is the one obtained with the change of variables of [3], [4] which requires full-order control, then to reduce (e.g. using balanced reduction) or augment (e.g. injecting new pole(s)) its size to the desired value. Remark that for full-order control with a single objective this change of variables leads to the global optimum, thus the technique proposed here is

useless in that case. But not anymore in the case of multi-objective control, requiring several Lyapunov matrices (this first change of variables can only consider one Lyapunov matrix). One can consider that the main dynamics needed by the controller can be found within this conservative full-order design, often used in the literature as initial solution which will also be the case here.

The method proposed here is, once the LMI optimization done and all the matrices obtained, to fix the S_2, T_{12}, T_{22} and to rewrite the optimization problem with the second set of variables: $v_\beta = A_K, B_K, C_K, D_K, R, S_1, T_{11}$. This builds a new convex subspace of the in general non-convex (especially in multi-objective with several performance channels, therefore several Lyapunov matrices) space of all solution. This new subspace is built around the first solution (or more precisely its S_2, T_{12}, T_{22} matrices). Therefore a new optimization of the objective can only lead to an improvement or at least to the same objective (the previous solution), since LMIs are solved globally efficiently thanks to Interior Point Methods [5], [6]. Writing Γ_α the objective obtained at the first step and Γ_β at the second, we have:

This second step, if it improved the objective, has changed the A_K, B_K matrices. Then the α step can be applied again around this new A_K, B_K matrices. If this again improves the objective ($\Gamma_\alpha < \Gamma_\beta$), we continue on with an other β step around the newfound S_2, T_{12}, T_{22} matrices and so on. This algorithm uses successive convex subspaces in order to dig deeper into the non-convex space of solutions. It can be seen as a block-coordinate algorithm, defining the coordinates: $x_\alpha = S_2, T_{12}, T_{22}$, $x_\delta = C_K, D_K, S_1, T_{11}$, $x_\beta = A_K, B_K$. This can only improve the solution, or at worst keep it unchanged.

An important remark is that this algorithm can be applied whatever the size of the control parameter K (thus fixed-order control). Any given controller K may thus be improved by this algorithm, provided that it is first (re)formulated (using e.g. [7]) as an observer-based controller (so that P as the desired structure of equation (1)) and that the objective to be improved can be written with LMIs.

The proof of convergence of this algorithm is ongoing work. For illustration purposes an example of convergence obtained is drawn below.

A convergence example

The example considered uses the same plant and control structure as that in [8]. Details are not reminded here, the results obtained are case-dependent. The arbitrary objective minimized here is a pondered sum of the \mathcal{H}_2 norm of the control sensitivity and the \mathcal{H}_∞ norm of the input sensitivity $\Gamma = \|S_U\|_2 + \lambda \|S_I\|_\infty$. The matrices A_K, B_K obtained with the full-order control change of variables (conservative because of the constraint of a unique Lyapunov matrix) are used as initial solution for the algorithm.

The examples of convergence are given in Fig. 2, for three different values of the trade-off parameter λ .

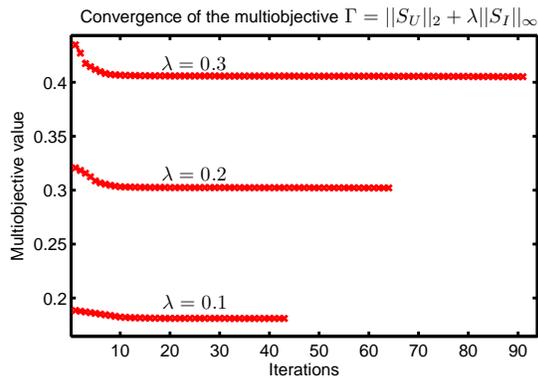


Fig. 2. Examples of convergence

Note that: -the stopping criterion chosen was when the objective decreased than less than 0.001% (thus the large number of iterations) -each iteration is made of the two steps α and β -the size of the controller has not been changed, thus kept here as full-order. The monotonous decrease of the objective can clearly be observed with these examples. The number of required iterations to reach a given precision varies from one objective to the next and can not be known beforehand (yet).

III. CONCLUSIONS

This work uses the approach of Linear Matrix Inequalities (LMIs) to study the problem of fixed-order output-feedback control design for LTI systems. Using Lyapunov theory and the Schur complement, many design objectives can be written within this frame. The difficulty of this problem is that the matrix inequalities at first involve products between variables: the Lyapunov matrices and the state-space matrices of the design parameter. Therefore these are Bilinear Matrix Inequalities for which the space of solutions can be non-convex and are well known to be hard to deal with.

A solution that has got a lot of attention in the last decade is to change the variables so that the inequalities become affine in the new variables, therefore turning in LMIs exploring convex subspaces of the in general non-convex space of all solutions. There exists two such changes of variables, the first one requires full-order control [4], [3], the second one fixed-order observer-based control [1], [2].

The contribution of this work is to make a new use of the second change of variables so that the non-convex space of all solutions is explored further, leading to improved objectives with reduced conservatism. This thanks to the key trick that the change of variables [1] can lead to two affine alternatives of the matrix inequalities. It is an alternate scheme method that can be qualified of block-coordinate descent algorithm since at each step a bloc of coordinates is fixed and the other is optimized. Note that in the studied

problem, using both sets at the same time is typically non-convex and is considered NP-hard in general.

Using the initial solution of full-order control with the first change of variables [3], [4] to capture the main dynamics needed by the fixed-order controller, and then applying this algorithm, is a new systematic solution where no *a priori* choice is needed.

The technique is successfully used with an illustrative example. This algorithm can be used to (try to) improve any given controller with respect to an objective that can be written as LMIs. Ongoing work studies the convergence of the proposed algorithm.

IV. ACKNOWLEDGMENTS

E. Simon and V. Wertz gratefully acknowledge the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization) funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

REFERENCES

- [1] C. Scherer, "An efficient solution to multi-objective control problem with lmi objectives," *Systems and Control Letters*, vol. 40, pp. 43–57, 2000.
- [2] B. Clement and G. Duc, "A multiobjective control via youla parameterization and lmi optimization: application to a flexible arm," in *Proc. of IFAC Symp. on Robust Control and Design, Prague, 2000*.
- [3] C. Scherer, P. Gahinet, and M. Chilali, "Multi-objective output-feedback control via lmi optimization," *IEEE Transactions on Automatic Control*, vol. 42, pp. 896–911, 1997.
- [4] I. Masubuchi, A. Ohara, and N. Suda., "Lmi-based controller synthesis: a unified formulation and solution," in *Proceedings of American Control Conference, 1995*.
- [5] Y. Nesterov and A. Nemirovski, *Interior point polynomial methods in convex programming : theory and applications*. SIAM Books, Philadelphia, 1994.
- [6] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization*. MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.
- [7] D. Alazard and P. Apkarian, "Exact observer-based structures for arbitrary compensators," *International Journal of Robust and Nonlinear Control*, vol. 9 Issue 2, pp. 101–118, 1999.
- [8] E. Simon, C. Stoica, P. Rodriguez-Ayerbe, D. Dumur, and V. Wertz, "Robustified optimal control of a coal-fired power plant," 2010, american Control Conference, Baltimore.