



**HAL**  
open science

## **LMIs-based coordinate descent method for solving BMIs in control design: A short survey & two improvements**

E. Simon, Pedro Rodriguez-Ayerbe, Cristina Stoica, Didier Dumur, Vincent Wertz

### ► **To cite this version:**

E. Simon, Pedro Rodriguez-Ayerbe, Cristina Stoica, Didier Dumur, Vincent Wertz. LMIs-based coordinate descent method for solving BMIs in control design: A short survey & two improvements. 18th IFAC World Congress, Aug 2011, Milan, Italy. 7 p. hal-00592350

**HAL Id: hal-00592350**

**<https://centralesupelec.hal.science/hal-00592350v1>**

Submitted on 16 Mar 2020

**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.

# LMIs-based coordinate descent method for solving BMIs in control design

E. Simon \* P. R-Ayerbe \*\* C. Stoica \*\* D. Dumur \*\* V. Wertz \*

\* *Department of Mathematical Engineering, Université Catholique de Louvain, 4 avenue Georges Lematre, 1348 Louvain-la-Neuve, Belgium*  
(e-mails {emile.simon\*; vincent.wertz}@uclouvain.be)

\*\* *SUPELEC Systems Sciences (E3S) Automatic Control Department, 3 rue Joliot Curie, 91192 Gif-sur-Yvette cedex, France*  
(e-mails {pedro.rodriguez; cristina.stoica; didier.dumur}@supelec.fr)

---

**Abstract:** This work investigates the topic of solving Bilinear Matrix Inequalities (BMIs) problems in the optimal control design field, using successive resolutions of properly defined Linear Matrix Inequalities (LMIs). This technique can be described as an ‘LMI-based coordinate descent method’. Indeed the original (BMI) problem is solved independently for each coordinate at each step using a LMI optimization, while the other coordinate is fixed. No method based on this idea has been formally proved to converge to the global optimum of the BMI problem, or even a local optimum. This will be discussed using relevant results both from the mathematical programming and control design points of view. This discussion supports the algorithm proposed here which, thanks to a particular change of variables, leads to sequences of improving solutions. Also emphasized is a second improvement important to avoid in practice early convergence to suboptimal solutions instead of local optima. The control framework used is that of optimal output feedback design for linear time invariant (LTI) systems. An example using a random plant is drawn to illustrate the typical effectiveness of the algorithm.

Keywords: Optimal control design, BMIs, LMIs, Output-feedback, LTI system

---

## 1. INTRODUCTION

This work is initially motivated by optimal control design objectives for which no global optimum is known, more specifically fixed-order or full-order output-feedback control design for multi-objectives problems with linear time invariant (LTI) systems. The aim is the reduction of conservatism of otherwise available solutions, ideally toward local optimality. Global optimality is typically NP-hard to achieve with such problems, and even the study of local optimization techniques in control design is not trivial and still open for investigation.

The approach considered is that of performing successive LMIs optimizations to solve locally BMIs. The motivation behind this idea is that LMIs can be efficiently solved using interior point methods (Nesterov and Nemirovski (1994), Ben-Tal and Nemirovski (2001)), unlike the available methods to solve BMIs which may only work for a few ‘complicating variables’ (Tuan and Apkarian (2000)). Two branches can be identified.

The first branch considers solving successive local LMIs approximations (like 1st order Taylor in (Ostertag (2009))) of the BMIs. These approximations only make sense in some unknown neighborhood of the considered local solution and each new solution has to be verified.

The second is to solve LMIs for which the feasible set is a subset of the original BMIs problem, thus leading to a probably conservative solution, then to use the obtained solution as an initial solution in a different subset of

the original BMIs problem and continue iteratively. Two kinds of subsets are considered: either the left or the right side variables of the bilinear terms are let free, and the others are fixed at their previously updated values. Thus the iterative procedure considers at each iteration two LMI optimizations, one for each BMI side. The advantage compared to the first branch is that each new solution is guaranteed to be a solution of the original problem (since each LMIs representation considered has a feasible set belonging to the BMIs feasible set). This technique can be classified as a (block-)coordinate descent algorithm. Both branches have methods that are not guaranteed to converge in general, globally or locally, although the new objective at each step must be better or the same than the previous objective. The work done here is confined to the second branch chosen because of its advantage that the successive solutions remain within the BMI feasible set. Nevertheless, the study of the first branch methods would partially lead to the same challenges and considerations than those made here.

The motivation behind this study is that such methods can very well work in practice even without formal proof of convergence. The approach is then to study these methods and propose improvements dealing with the difficulties encountered. This paper’s contributions are: the short survey of such methods, the explanations of the improvement introduced by the considered change of variables and the second improvement that can be used to avoid premature convergence of this kind of algorithms. In the end the

proposed improved algorithm will be shown to perform very well in practice, although it still lacks a formal proof of convergence.

This work is structured as follows. The involvement of BMIs in control design is reminded, along with notations (Section 2). The algorithm used in this work is described (Section 3). Following a short survey of the literature (Subsection 4.1), the first improvement brought by the is highlighted (Subsection 4.2). The second improvement is given, that is actually not intrinsic to the proposed algorithm but can be applied to any LMI-based coordinate method for BMIs (Subsection 4.3). Then a random example illustrates the proposed algorithm and improvement effectiveness (Section 5). Finally, the conclusions are drawn and future work directions are listed.

## 2. BMIS IN CONTROL DESIGN

We use classical  $P$ - $K$  representation and notations:

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

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

Stabilizing controllers exist if  $(A; B)$  is stabilizable and  $(A; C)$  is detectable, which is assumed from now on. The closed-loop performance channel  $T_{ij} : w_i \rightarrow z_j$  is obtained:

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

The  $\mathcal{H}_2$  norm of a continuous time performance transfer function  $T(s)_{ij}$ :  $\|T(s)_{ij}\|_2 < \gamma_{ij}$  iff  $\text{trace}(\mathbf{Z}_{ij}) < \gamma_{ij}$  and

$$\begin{pmatrix} \mathbf{A}^T \mathbf{X}_{ij} + \mathbf{X}_{ij} \mathbf{A} & \mathbf{X}_{ij} \mathbf{B}_i \\ * & -\gamma_{ij} I \end{pmatrix} < 0, \begin{pmatrix} \mathbf{X}_{ij} & \mathbf{C}_j^T \\ * & \mathbf{Z}_{ij} \end{pmatrix} \succ 0, \mathbf{D}_{ij} = 0$$

The optimization variables are the following:

- The Lyapunov matrices  $\mathbf{X}_{ij} = \mathbf{X}_{(ij)}^T \succ 0$  associated to their channels  $T_{ij}$
- The state-space matrices  $A_K, B_K, C_K, D_K$
- The objectives variables  $\gamma_{ij}$  positive scalars,  $\mathbf{Z}_{ij}$  symmetric positive definite matrices

The notation  $*$  indicates symmetrical terms. Note that these matrices are BMIs, because of bilinear terms consisting in the product of the Lyapunov matrices and the design parameter state-space matrices.

No method is known to work well to solve BMI problems, except maybe for small problems with few ‘complicating’ variables (Tuan and Apkarian (2000)). It is much more suitable to use LMIs reformulations, efficiently solved using interior point methods (Nesterov and Nemirovski (1994); Ben-Tal and Nemirovski (2001)). For the LTI systems output-feedback control design problem considered here, there exist two changes of variables turning the BMIs in LMIs. The first method (Masubuchi et al. (1995)) requires full-order control and the second is method (Scherer (2000)) requires observer-based control, both allowing to overcome the difficulty of products between Lyapunov

matrices and the state-space matrices of the design parameter. Using these, convex subspaces (feasible sets of LMIs) of the in general non-convex space of all solutions are considered.

## 3. PROPOSED ALGORITHM

### 3.1 The change of variables

The algorithm is given after the description of the change of variables it uses. The state-space matrices of  $P$  must have the following structure, necessary for the considered change of variables:

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

The previous notations are close to that of (Scherer (2000)). The states of  $P$  are divided in two parts: the first is the difference between the actual plant and observer states  $\epsilon$  and the second is the rest of the states  $\chi$ . This notation is chosen according to the use of a structure with observer, in which case the input  $y_P$  of the design parameter  $K$  is the difference between the actual plant output  $y$  and the observer output  $\hat{y}$ . Note that with the structure (1),  $P$  is stable due to the structure with observer. If a given  $P$  is not structured as (1), the more general way to enforce this structural property is to use the Youla parameterization to span the set of all stabilizing controllers (as reminded in (Scherer (2000))). Notice that this doubles the size (number of states) of the original plant  $P$ .

The particularity of such structure is that the transfer between  $y_P$  and  $u_P$  is zero, consequently the transfer between  $w$  and  $z$  is affine on the  $K$  parameter. Then the control parameter can be inserted within the channel of  $P$  with output  $y_P$  and input  $u_P$  respecting this structural property:  $T_{u_P y_P} = C(sI - A)^{-1} B = 0$  (Scherer (2000)).

Using the change of variables of (Scherer (2000)), the BMIs are rewritten as follows:  $\mathbf{X} \rightarrow \mathbf{X}(v)$ ,  $\mathbf{X}\mathbf{A} \rightarrow \mathbf{A}(v)$ ,  $\mathbf{X}\mathbf{B}_i \rightarrow \mathbf{B}_i(v)$ ,  $\mathbf{C}_j \rightarrow \mathbf{C}_j(v)$  where each of these now affine terms is defined hereunder.

The term  $\mathbf{D}_{ij} = D_{ij} + E_j D_K F_i$  does not change.

The Lyapunov matrices are decomposed and changed into  $\mathbf{X}_{ij} \rightarrow (R_{ij}, S_{1ij}, S_{2ij}, T_{11ij}, T_{12ij}, T_{22ij}) = v$  detailed in (Scherer (2000)), (Stoica et al. (2007)). In order to lighten the notations, the  $ij$  indices (denoting the considered performance channel and its associated Lyapunov matrix) are omitted in the following terms:

$$\begin{aligned} \mathbf{A}(v) &= \begin{pmatrix} A_1 R & t_1 & t_2 \\ 0 & T_{11} A_2 + T_{12} B_K C_{\epsilon y_P} & T_{12} A_K \\ 0 & T_{12}^T A_2 + T_{22} B_K C_{\epsilon y_P} & T_{22} A_K \end{pmatrix} \\ \mathbf{B}(v) &= \begin{pmatrix} B_{u_P \chi} D_K D_{w y_P} + B_{w \chi} - S_1 B_{w \epsilon} - S_2 B_K D_{w y} \\ T_{11} B_{w \epsilon} + T_{12} B_K D_{w y} \\ T_{12}^T B_{w \epsilon} + T_{22} B_K D_{w y} \end{pmatrix}, \end{aligned}$$

$$\mathbf{C}(v)^T = \begin{pmatrix} RC_{xz}^T & & \\ & t_3 & \\ S_2^T C_{xz}^T - C_K^T D_{uz}^T & & \end{pmatrix}, \mathbf{X}(v) = \begin{pmatrix} R & 0 & 0 \\ * & T_{11} & T_{12} \\ * & * & T_{22} \end{pmatrix}$$

$$t_1 = A_1 S_1 - S_1 A_2 - S_2 B_K C_{\epsilon_{yP}} + A_3 + B_{uP\chi} D_K C_{\epsilon_{yP}}$$

$$t_2 = A_1 S_2 - S_2 A_K + B_{uP\chi} C_K$$

$$t_3 = S_1^T C_{xz}^T + C_{\epsilon z}^T - C_{\epsilon_{yP}}^T D_K^T D_{uz}^T$$

This change of variables has only been used fixing the  $A_K, B_K$  matrices beforehand. Indeed, this *a priori* choice is necessary since  $A_K, B_K$  appear in product terms.

Of course this arbitrary choice of  $A_K, B_K$  is determinant to reach the objectives. This choice has so far only been done using a polynomial expansion defining a basis of the space of all controllers (Scherer (2000)). By increasing the size of the polynomial expansion, thus that of the  $A_K, B_K$  matrices and therefore of the LMIs to be solved, the objective converges to the global optimum. However a very large size may be required to obtain such convergence, that is when the optimal dynamics (poles) of the optimal controller appear only at high orders of the expansion. This is why this arbitrary choice is determinant for the convergence of that method or even whether it will be able to provide ‘good enough’ (i.e. not too far from the optimum) controllers for a size realistic in practice.

An other idea is developed here, coming from the following key observation: only the variables  $S_2, T_{12}, T_{22}$ , related to the Lyapunov matrices, multiply the  $A_K, B_K$ .

### 3.2 The proposed algorithm

Following the key observation, the new variables can be regrouped in the following three (block-)coordinates:  $x_\alpha = S_2, T_{12}, T_{22}$ ,  $x_\delta = C_K, D_K, R, S_1, T_{11}$ ,  $x_\beta = A_K, B_K$ . Then, considering  $x_\delta$  always appears affinely with the change of variables, the implementation of the following algorithm is done quite logically:

*The new LMI-based coordinate descent algorithm:*

- 1) Choose an initial solution (e.g. three choices hereunder)
- 2) Fix variables  $x_\beta$  then optimize the objective using LMI optimization with  $v_\alpha = x_\alpha, x_\delta$  as free variables
- 3) Fix variables  $x_\alpha$  then optimize the objective using LMI optimization with  $v_\beta = x_\delta, x_\beta$  as free variables
- 4) Iterate successively both steps 2) & 3), fixing at each step  $x_\beta$  or  $x_\alpha$  previously updated, until the objective decreases less than a chosen accuracy.

The proposed algorithm allows to further explore the non-convex space of all solutions using successive convex subspaces. This is illustrated in Figure 1, drawing a general representation of the algorithm first few steps.

For convenience, three possible choices are pointed out for the initial solution. The first is to use any random stable controller, or rather several to seek several local optima (note that if these optima are the same, this is an indication these might be the global optimum but this is an other topic). The second is to start from any observer-based controller otherwise available, or to turn any output-feedback control into an observer-based structure (this can be done exactly but implicitly using (Alazard and Apkarian (1999))). The third is to design a full-order controller using the other change of variable

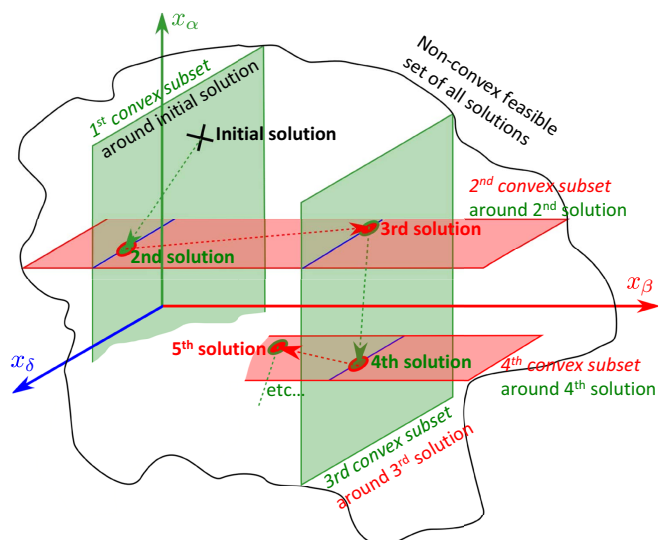


Fig. 1. Representation of proposed algorithm

available (Masubuchi et al. (1995)). When more than one objective functions are considered, this solution bears the conservatism of a unique Lyapunov matrix. Note that if a smaller size is desired, this initial solution can be reduced using e.g. balanced reduction.

## 4. LMI-BASED COORDINATE DESCENT METHOD

The algorithm now presented, a short survey of the literature is made to present relevant elements regarding the convergence of such techniques. It seems appropriate to structure the existing approaches in the literature in two parts. The first and more general part concerns the mathematical programming about coordinate descent methods. The second part deals with the control design where such methods are applied towards objective optimization. At several places in control design literature it can be read that coordinate descent methods are not guaranteed to converge (Kanev et al. (2004); Iwasaki (1999); Yamada and Hara (1998)). However no explanations are given, although such methods can only improve or maintain the objective. This is the motivation behind the next subsection, presenting the few central elements found within each community.

### 4.1 Short survey of literature

**Mathematical programming elements** The idea of fixing a set of variables while optimizing the remaining free set of variables and iterating for each set of variables is called (block-)coordinate descent method, meaning that the cost is minimized along one (block-)coordinate direction at each iteration. The methods currently considered fits this general definition, thus the designation of LMI-based coordinate descent methods. Therefore the related central result of (Bertsekas (1999)) is reminded: every limit point of such method is a stationary point under a few requirements. A first requirement is that the cost function has to be continuously differentiable. In the current context, this is always met because the objective functions (minimized under LMI constraints with interior point methods) have to be linear combinations of the decision variables. A second hypothesis is that the set of solutions  $X \subseteq \mathbb{R}^n$  must be structured as a Cartesian product of

closed, nonempty and convex subsets  $X_i \subseteq \mathbb{R}^{n_i}$ . Because of the in general non-convex feasible sets of BMIs, this result cannot be used here.

A very relevant paper is that of Helton and Merino (1997) considering the optimization of the largest eigenvalue of a smooth selfadjoint matrix valued function  $\Gamma(X, Y)$ , which is bi-convex (convex in  $X$  and  $Y$  separately). The problem of minimizing this objective can be written as:

$$\min t \text{ s.t. } \text{BMIs}(X, Y) < tI \quad (2)$$

where  $t$  is a scalar and BMIs depend on two matrix variables  $X$  and  $Y$ . This objective is not exactly the more general objective of minimizing a linear combination of decision variables but it is similar. Actually such objective can be turned into the feasibility of an LMI, if the desired objective values are known beforehand. Anyhow, the paper of Helton and Merino (1997) provides strong evidence (but not a formal proof) that the technique of minimizing the largest eigenvalue by updating alternatively  $X$  and  $Y$  will almost never reach a local solution. This gives an important result to identify the reason why the general method studied does not converge to a local optimum. From this result is drawn the first hint that will lead to the second improvement presented in subsection 4.3.

In the end, the difficulty stems from the BMI constraints and their non-convex feasible set, leading to NP-hard problems. This feature blocks the development of formal proof of convergence for most methods, including LMI-based coordinate descent.

**Control design elements** In (Goh et al. (1995)), a counter-example is given for which the method (called ‘alternating LMI for BMI’) does not converge to a local solution. This echoes the result from Wendell and Hurter (1976) in mathematical programming, where we read that such method leads to ‘partial optimal’ solutions that may not be local optima. Therefore in general the convergence of the considered methods can not be proved.

In the paper of (Iwasaki (1999)), it is reminded that this kind of algorithms generates a monotonically nonincreasing sequence but are not guaranteed to converge to the global optimum. It is also said that the efficiency and reliability of the algorithm depends critically on the choice of coordinates (and not only on the initial solution, always important for local optimization).

In (El Ghaoui and Balakrishnan (1994)), the first algorithm falling within the current scope is proposed. It can be read in this paper that locally optimal solutions are obtained, but the proof of convergence still has to be done. The method proposed there is straightforward: since BMIs in control design come from products between two sets of decision variables: the Lyapunov matrix(ces)  $V$  and the controller state-spaces matrices  $K$ , then LMIs having either one of both sets as decision variables can be solved instead. Thus, starting from an initial solution, the objective can be iteratively optimized alternating at each step between the two sets of variables  $V$  and  $K$ : the so-called  $V$ - $K$  iteration. This algorithm is further discussed hereunder.

Also noted are the works of Banjerdpongchai and How (1997) considering a very similar algorithm to the one

proposed in Section 3, but with another change of variables, used for Popov  $\mathcal{H}_\infty$  or  $\mathcal{H}_2$  robustness of parametric uncertain systems. They came around observations that back up ours and some of these are reminded in the next subsection.

#### 4.2 First improvement

This subsection develops the reflexions and observations explaining why the proposed algorithm is expected to perform well. In particular, this development will be made by putting the emphasis on the reasons why the  $V$ - $K$  iteration should not work in practice and what is then the improvement brought by the new algorithm. A sufficient condition to check the local optimality of a solution is also reminded here.

The straightforward idea of the  $V$ - $K$  iteration (El Ghaoui and Balakrishnan (1994)) carries an intrinsic weakness that often leads to poor or no convergence at all. Control design is about changing the properties of a system by inserting and designing a controller. In optimal control design, this is done through the optimization of a mathematical criterion regarding the behavior or properties of the controlled system. When the controller is completely fixed, the entire system representation is fixed and none of its properties can be changed. Therefore, at the  $V$ -update step of the  $V$ - $K$  iteration no closed-loop improvement can be made and the actual control design objective value remains identical to the previous value found at the end of the  $K$ -update step. Alternatively, any change in the objective obtained there is not representative of any change in the system. This method could only work if at the  $V$ -update step the Lyapunov matrix is modified so that it becomes ‘oriented’ in such a way that, at the following  $K$ -update step, this new Lyapunov matrix fits for a better controller. Nothing ensures such a special property. Considering this, the convergence of the  $V$ - $K$  iteration toward local optimality can most probably not be guaranteed.

The algorithm proposed in Section 3 bears a big difference compared to the obvious  $V$ - $K$  iteration, thanks to the change of variables: at each step either all of the controller’s matrices or at least the  $C_K, D_K$  matrices are let free as decision variables. Thus, the closed-loop can change and the actual objective’s value can be improved at each step. Compared to the  $V$ - $K$  iteration and the general negative results from the literature, this algorithm has three sets of variables instead of two, one of which is always free. These arguments from the two perspectives (control design and mathematical programming) designate the improvement of the proposed algorithm over the  $V$ - $K$  iteration. This can be reformulated for both points of view as follows. From control design, an improvement can be brought at each step through the  $C_K, D_K$  controller matrices. From mathematical programming, the bilinear terms now encountered are written in general as  $x_\alpha x_\beta + x_\delta$ , thus where  $x_\delta$  appears as an additional affine degree of freedom making a possible ‘blocking’ of the algorithm much less likely.

This important difference was also highlighted by D. Banjerdpongchai (1998) mentioning these ‘shared variables’ in the LMIs, actually offering an additional degree of freedom

which is a significant improvement compared to the original  $V$ - $K$  iteration. However still no proof of convergence is given (only an a-posteriori exhaustive search verification). In the previous article of Banjerdpongchai and How (1997) is reminded that the convergence is implied under the conjecture of the objective being reduced at each step. It is also noted that on more complex objectives like  $\mathcal{H}_2$  norm of performance channels, the  $V$ - $K$  iteration converges very slowly or even not at all.

The to prove this general convergence criterion: this algorithm converges if and only if the search directions considered at each iteration bear an improved solution, until the solution is locally optimal. This is more likely with the algorithm proposed here (thanks to the ‘shared variables’ always free at each step, among which the  $C_K, D_K$  matrices of the controller’s state space representation) but is still not proved because of the BMIs non-convex feasible sets.

On the other hand what can be guaranteed is the side result that under the following weak assumption: (H1) *The sub-optimal initial controller  $K$  has state-space matrices  $C_K, D_K$  not optimal (with respect to the considered objective) for  $A_K, B_K$  then at least the first step of the algorithm improves the objective. Indeed this step is guaranteed to find the globally optimal  $C_K, D_K$  for fixed  $A_K, B_K$  controller’s state-space matrices (Scherer (2000)). In practice (H1) is always verified.*

The classical sufficient condition to check the local optimality of a given solution is reminded here. It is to verify whether there may still exist directions in which the objective could be improved. This is made by checking whether a ‘local virtual objective’ is better (by at least some given accuracy) than the actual solution objective. This ‘virtual objective’ is obtained using the first order approximation of the BMIs around the given solution, which are LMIs. These LMIs constraints are weaker than the BMIs constraints and minimizing the objective under these LMIs will provide the virtual objective which is a lower bound for the actual objective, locally. Thus if the actual solution’s objective is close (to a given accuracy) to its virtual objective, it is a local optimum. Otherwise it still could be (depending on the limits of the BMI feasible set), but this cannot be verified with this condition. The LMIs approximation of the BMIs around the current solution  $\bar{x}_\alpha, \bar{x}_\beta$  is written with  $x_\alpha = \bar{x}_\alpha + \Delta x_\alpha, x_\beta = \bar{x}_\beta + \Delta x_\beta$ :  
 $\text{BMIs}(x_\alpha, x_\beta, x_\delta)|_{\bar{x}_\alpha, \bar{x}_\beta} \cong$   
 $\text{LMIs}(\bar{x}_\alpha, \bar{x}_\beta + \bar{x}_\alpha \Delta x_\beta + \Delta x_\alpha \bar{x}_\beta + \Delta x_\alpha \Delta x_\beta, x_\delta)$   
 so the local bilinear term  $\Delta v_\alpha \Delta v_\beta$  has been relaxed.

### 4.3 Second improvement

Current solvers could be easily extended to alternate through two sets of variables properly, thus effectively implementing the local BMI resolution with LMI-based coordinate descent. This would provide a convenient first attempt to solve BMI problems. Then here is highlighted a very important feature that should not be forgotten: the choice of the alternating criterion determining the moment when the current set of variables must be frozen and the other freed. To the best of the authors’ knowledge it appears that so far LMIs-based coordinate descent algorithms to solve BMIs only alternate the variables

at the normal end of each LMI optimization: at each coordinate update, the considered objective is minimized to the solver’s accuracy. Two reasons made us notice this criterion: the results in (Helton and Merino (1997)) and the general convergence criterion reminded in the previous subsection.

So far this method computes the best coordinate for the other fixed coordinate toward the considered objective at each step (the global optimum of this sub-problem obtained with LMI optimization). Therefore with this ‘hidden’ or implicit alternating criterion, at each step these are actually local sub-problems (find the best coordinate for the other fixed) that are solved and not exactly the original BMI problem anymore. In practice what may happen is the situation of partial optimal solution that are not local optima, as announced in general in (Wendell and Hurter (1976)) and in particular in (Helton and Merino (1997)). This can be interpreted as a bad ‘localization’ of the algorithm: the considered solutions enter a sequence pertaining to this sub-problems scheme but not anymore to the global problem.

Instead, the alternating decision should be based on the general convergence criterion: the method converges to a local optimum if a better solution is obtained at each iteration. Thus what is needed is to *compute one better solution at each step, and not necessarily the best*. So, what is only required is to find a feasible solution to the problem that has a better objective than the previous one. This can be written as solving at each step the following feasibility problem:

$$feasp\{\text{LMI}_{i+1} + (\Gamma_{i+1} < \Gamma_i - \epsilon)\} \quad (3)$$

where *feasp* means to find one solution respecting the given constraints,  $\text{LMI}_{i+1}$  are the LMI constraints at the next step,  $\Gamma_{i+1}$  the objective at the next step and  $\epsilon$  a chosen accuracy. Thus, the bad ‘localization’ of the algorithm is much more unlikely: under this scheme the improvement of the original problem’s objective is done more simultaneously in both variables than with the usual ‘the best for the other’ alternating criterion.

In our experience this works fine using  $\epsilon = 0$  because for the solver  $\Gamma_{i+1} < \Gamma_i$  is only just another constraint and it will not necessarily behave so that  $\Gamma_{i+1}$  gets stuck just under  $\Gamma_i$  at each step. On a practical note, it is suggested to use the very robust *feasp* algorithm of the *LMILab* toolbox in *Matlab*. Note also that with this method, only feasibility (‘feasp’) computations of constraints are made which is significantly faster than to minimize an objective under these constraints.

Finally, we remark that the article of Helton and Merino (1997) provides strong evidence that: on one hand for the MIN-MIN problem (minimization in both ‘sides’ of the BMI) the coordinate optimization almost never reaches a local solution but on the other hand for the MIN-MAX (minimization in one side and maximization in the other) it almost always does. The adaptation of these results to show that the proposed ‘feasp-feasp’ method converges is currently studied. Likewise this new alternating rule is expected to keep the intermediate solutions away from the BMI limits: the possible interpretation as a ‘BMI interior point method’ is also under further investigation.

## 5. RANDOM EXAMPLE

The objective is to minimize the  $\mathcal{H}_2$  norm of the single performance channel of a random plant. The discrete-time random plant considered is given by (already including a Youla parameterization enforcing structure (1)):

$$A = \left( \begin{array}{ccc|ccc} -1.41 & -0.008615 & 0.9433 & 5.642 & 0.5387 & -3.05 \\ 2.519 & 0.08446 & -1.605 & 8.342 & 1.068 & -4.446 \\ -2.29 & -0.01333 & 1.53 & 6.336 & 0.9779 & -3.337 \\ \hline 0 & 0 & 0 & 2.088 & -0.5969 & 0.04967 \\ 0 & 0 & 0 & 5.827 & -1.641 & 0.1544 \\ 0 & 0 & 0 & 1.262 & -0.4047 & 0.2607 \end{array} \right),$$

$$B_1 = \begin{pmatrix} 0.9218 & 0.4057 \\ 0.7382 & 0.9355 \\ 0.1763 & 0.9169 \\ \hline 0.9218 & 0.4057 \\ 0.7382 & 0.9355 \\ 0.1763 & 0.9169 \end{pmatrix}, \quad B = \begin{pmatrix} 0.9218 & 0.4057 \\ 0.7382 & 0.9355 \\ 0.1763 & 0.9169 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},$$

$$C_1 = \left( \begin{array}{ccc|ccc} 0.4103 & 0.05789 & 0.8132 & 0 & 0 & 0 \\ 0.8936 & 0.3529 & 0.009861 & 0 & 0 & 0 \end{array} \right), \quad D_{11} = 0$$

$$C = \left( \begin{array}{ccc|ccc} 0 & 0 & 0 & 0.4103 & 0.05789 & 0.8132 \\ 0 & 0 & 0 & 0.8936 & 0.3529 & 0.009861 \end{array} \right), \quad E_1 = 0, F_1 = 0.$$

Fig. 2 provides the algorithms' results.

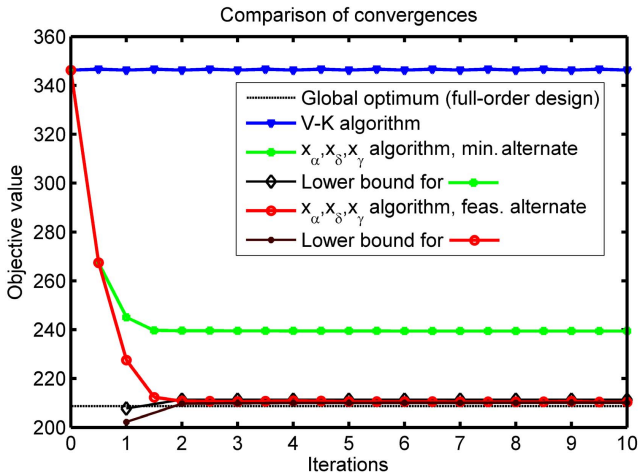


Fig. 2. Example of convergences

The horizontal line gives the global optimum, obtained with the full-order (order 6) change of variables from (Masubuchi et al. (1995)). Note that this global optimum for all controllers is available because only a single objective is considered here, thus the unique Lyapunov matrix available with this change of variables is sufficient. Then this full-order controller is reduced from 6 to 1 state using balanced truncation (note that only proper controllers have been considered, thus with  $D_K = 0$ ).

This gives the initial solution from which the other algorithms are started (and with the associated objective value given at the iteration 0). Note that for this reduced-order design the global optimum is not known.

The first result is obtained with the  $V$ - $K$  algorithm. As expected, no improvement is obtained and the objective remains at the same value (this was obtained without the constraint  $\Gamma_{i+1} < \Gamma_i$ , otherwise that algorithm breaks immediately in infeasibility).

The second result is obtained with the proposed algorithm with the implicit ‘min’ alternating rule (the objective is minimized at each step). A much better convergence is observed than that of the  $V$ - $K$  algorithm, thanks to the  $x_\delta$  variables free at each step. However that solution does not verify the sufficient local optimality condition (its local lower bound is far below, under the bottom curves) and we can not say whether or not this is really a local optimum.

The third and best result is obtained with the proposed algorithm with the improved ‘feas’ alternating rule (*one better* solution is obtained at each step). As can be seen on the figure, the local lower bound clearly converges toward the same value. Therefore the algorithm has reached a local optimum. Besides, the global optimum for all controllers is quite close so it is very much possible that this local optimum is actually global for this reduced design, though this can not be proved.

## 6. CONCLUSIONS

The approach of this paper is a study of the convergence properties of algorithms that can be described as LMIs-based coordinate descent method to solve BMIs locally (from an initial solution). The framework considered is control design, from which originate the first and most of BMIs problems, and more specifically output-feedback optimal control for LTI systems. After a description of a new algorithm fitting to tackle such problems, a short survey of the literature is made to present relevant elements regarding the convergence of these algorithms in general or the one proposed in particular. These allow to shed light on the reasons why the proposed algorithm bears a first improvement compared to the other algorithm available, the  $V$ - $K$  iteration. Following this, a second important improvement that can be applied to the considered algorithms is given. Finally, an illustration of the discussed algorithms and improvements efficiency is made using a random example.

As it appears, the convergence of such method can not be proved in general. Indeed, although the objective can only become better or remain the same at each step, a proof of convergence would require that the feasible LMI directions (subspaces of the original BMI feasible set) considered at each iteration always bear an improved solution, until local optimality is reached. This is *a priori* not guaranteed for the proposed algorithm, the  $V$ - $K$  iteration or the method in general. Modifications to ensure this are not straightforward because of the inherent’s problem difficulty: the BMI constraints and their in general non-convex feasible sets (typically leading to NP-hard problems).

On the other hand the algorithm proposed bears a first improvement being the property that a set of variables remain free at each step, among which the  $C_K, D_K$  matrices of the controller’s state-space representation. Although it is not formally guaranteed, it is explained and observed that these variables lead to sequences of improving solu-

tions. The other main contribution is the second improvement that should always result in a better behavior for the algorithm proposed and in general, not only for the convergence but also for the computation time at each step. This is made with a discussion on when to alternate from one set of variables to the other, a feature apparently overlooked so far with such algorithms.

In the end the improved algorithm, although yet lacking a formal convergence proof, performs very well in practice. This has been illustrated with only one random example here but the same performance and positive results were obtained with all the other tests we tried so far. It is suggested that actual LMI solvers could be easily extended with the improved method to feature this easy systematic first attempt at solving BMI problems. Of course it might very well not bring any improvement, but it still remains a useful tool to be used in addition or combination with other methods.

Future works will study the combination of the presented algorithm with other optimization techniques to improve efficiency and or try to enforce convergence properties. In that sense, we will study further the improved alternating criterion and handling of successive solutions that may give an improved efficiency and/or convergence. Likewise we will look into possible new or improved search directions that would prevent the algorithm of getting stuck before reaching local optimality, possibly not only using coordinate descent but also linear approximations of the BMIs. Of course this will be done along with a study of the first branch of methods described in the introduction. The effectiveness will also be illustrated using extensive numerical experiments, considering both random systems and typical benchmark systems used in the literature along with mixed objectives. A `Matlab` function implementing the proposed algorithm can be written, and perhaps an extension of the `LMILab` toolbox for the systematic BMI approach suggested. Finally the technique might be enlarged in other frameworks like branch-and-bound or swarm optimization to deal with global optimization.

#### ACKNOWLEDGEMENTS

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

#### REFERENCES

- D. Alazard and P. Apkarian. Exact observer-based structures for arbitrary compensators. *Int. J. Robust Nonlinear Control*, vol.9(2) pp 101–118, 1999.
- D. Banjerdpongchai and J. P. How. Convergence analysis of a parametric robust  $\mathcal{H}_2$  controller synthesis algorithm. In *Proc. IEEE CDC*, pp 1020–1025, 1997.
- A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization*. MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.
- D. P. Bertsekas. *Nonlinear Programming: Second Edition*. Athena Scientific, Massachusetts, 1999.
- J.P. How D. Banjerdpongchai. Parametric robust  $\mathcal{H}_\infty$  controller synthesis: comparison and convergence analysis. In *Proc. IEEE ACC*, pp 2403–2404, 1998.
- L. El Ghaoui and V. Balakrishnan. Synthesis of fixed-structure via numerical optimization. In *Proc. IEEE CDC*, pp 2678–2683, 1994.
- K-C. Goh, M. G. Safonov, and G. P. Papavassilopoulos. Global optimization for the biaffine matrix inequality problem. *J. Global Opt.*, vol.7 pp 365–380, 1995.
- J. W. Helton and O. Merino. Coordinate optimization for bi-convex matrix inequalities. In *Proc. IEEE CDC*, pp 3609–3613, 1997.
- T. Iwasaki. The dual iteration for fixed-order control. *IEEE TAC*, vol.44(4) pp 783–788, 1999.
- S. Kanev, C. Scherer, M. Verhaegen, and B. De Schutter. Robust output-feedback controller design via local BMI optimization. *Automatica*, vol.40 pp 1115–1127, 2004.
- I. Masubuchi, A. Ohara, and N. Suda. LMI-based controller synthesis: a unified formulation and solution. *Int. J. Robust Nonlinear Control* vol.8 pp 669–686, 1998
- Y. Nesterov and A. Nemirovski. *Interior point polynomial methods in convex programming : theory and applications*. SIAM Books, Philadelphia, 1994.
- E. Ostertag. Less conservative  $\mathcal{H}_2/\mathcal{H}_\infty$  design of dynamic output-feedback controllers via LMI's. In *ECC*, 2009.
- C.W. Scherer. An efficient solution to multi-objective control problem with LMI objectives. *Systems & Control Letters*, vol.40 pp 43–57, 2000.
- C. Stoica, P. Rodríguez-Ayerbe, and D. Dumur. Off-line improvement of multivariable model predictive control robustness. In *Proc. of IEEE CDC*, 2007.
- H. Tuan and P. Apkarian. Low nonconvexity-rank bilinear matrix inequalities: Algorithms and applications in robust controller and structure designs. *IEEE TAC*, vol.45(11) pp 2111–2117, 2000.
- R.E. Wendell and A.P. Hurter. Minimization of a non-separable objective function subject to disjoint constraints. *Operation Res.*, vol.24(4): pp 643–657, 1976.
- Y. Yamada and S. Hara. Global optimization for  $\mathcal{H}_\infty$  control with constant diagonal scaling. *IEEE TAC*, vol.43 pp 191–203, 1998.