Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

APPLIED MATHEMATICS AND COMPUTATION

ELSEVIER Applied Mathematics and Computation 164 (2005) 531–543

www.elsevier.com/locate/amc

Control of large-scale systems in a multiprocessor environment

A.I. Zečević, D.D. Šiljak *

Department of Electrical Engineering, Santa Clara University, Santa Clara, CA 95053-0569, USA

Abstract

In this paper, a new approach is proposed for designing robust controllers for large-scale systems. The method utilizes linear matrix inequalities (LMI) to produce control structures that are suitable for a multiprocessor environment. It is shown that appropriate gain matrices can be obtained with only a modest computational effort, and that the interprocessor communication can be minimized.

Keywords: Control of large-scale systems; Linear matrix inequalities; Robustness; Parallel processing

1. Introduction

The problem of designing robust control for large-scale systems has received considerable attention over the past few decades. The literature on this subject is very extensive, and includes a number of comprehensive surveys (see [1,2] and the references therein). Although existing design strategies vary widely, it is fair to say that they all need to address the following basic requirements:

* Corresponding author.
E-mail address: dsiljak@scu.edu (D.D. Šiljak).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2004.06.032

(i) The determination of a stabilizing gain matrix should not require excessive computation.
(ii) The control law must incorporate any inherent information structure constraints.
(iii) The feedback must be easy to implement.
(iv) The control law needs to be robust with respect to modeling and parametric uncertainties.

Satisfying all of these objectives simultaneously is a major theoretical and practical challenge. Much of the work related to this problem has focused on *decentralized* control schemes, since they require only locally available information, and can be easily implemented in a multiprocessor environment. For such a design, the system is typically represented as a collection of N interconnected subsystems

$$\dot{x}_i = g_i(x_i) + h_i(x) + B_i u_i \quad (i = 1, 2, \dots, N) \tag{1}$$

and the feedback law takes the form

$$u_i(x) = K_i x_i. \tag{2}$$

From the computational standpoint, the simplest approach for designing decentralized control is to first stabilize the decoupled closed-loop subsystems

$$\dot{x}_i = g_i(x_i) + B_i K_i x_i. \tag{3}$$

The stability of the overall system and the corresponding degree of robustness with respect to uncertainties can then be established using vector Lyapunov functions (e.g. [1–3]). This method was found to be particularly effective in cases when the subsystems are weakly coupled (a property that can be detected using epsilon decomposition [4–6]). Alternatively, the design of decentralized control can be based on Linear Matrix Inequalities (LMI) and convex optimization [7–10]. In that case the robustness bounds are less conservative, but the computation is considerably more complex.

One of the limiting factors associated with decentralized control is the lack of information exchange between the subsystems. Allowing for such an exchange would clearly improve the system performance, since the control would be based on a larger information set. On the other hand, this type of design introduces a number of new problems, such as the identification of suitable structures for the gain matrix, and the minimization of interprocessor communications. With that in mind, the main objective of this paper will be to develop efficient control strategies that are suitable for a *multiprocessor environment*. In all cases the design will be based on linear matrix inequalities, using the mathematical framework proposed in [11–13] as a starting point. In the following sections, we will establish that this approach can accommodate several generic types of information exchange, while minimizing interprocessor communications.

2. Control design with information exchange between preassigned subsystems

Let us consider a nonlinear system described by the differential equations

$$\begin{aligned} \dot{x} &= Ax + h(x) + Bu, \\ y &= Cx, \end{aligned} \tag{4}$$

where $x \in R^n$ is the state of the system, $u \in R^m$ is the input vector and $y \in R^q$ is the output. A, B and C are constant $n \times n, n \times m$ and $q \times n$ matrices, and $h: R^n \to R^n$ represents a piecewise-continuous nonlinear function satisfying $h(0) = 0$. It is assumed that the term $h(x)$ can be bounded by a quadratic inequality

$$h^{\mathrm{T}}(x)h(x) \leqslant \alpha^2 x^{\mathrm{T}} H^{\mathrm{T}} H x, \tag{5}$$

where H is a constant matrix, and $\alpha > 0$ is a scalar parameter.

Given a linear feedback control law $u = Kx$, the global asymptotic stability of the closed-loop system can be established using a Lyapunov function

$$V(x) = x^{\mathrm{T}} P x, \tag{6}$$

where P is a symmetric positive definite matrix. Sufficient conditions for stability are well known, and can be expressed as a pair of inequalities

$$\begin{aligned} & x^{\mathrm{T}} P x > 0, \\ & \begin{bmatrix} x \\ h \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} (A+BK)^{\mathrm{T}} P + P(A+BK) & P \\ P & 0 \end{bmatrix} \begin{bmatrix} x \\ h \end{bmatrix} < 0, \end{aligned} \tag{7}$$

Defining $Y = \tau P^{-1}$ (where τ is a positive scalar), $L = KY$, and $\gamma = 1/\alpha^2$, the control design can now be formulated as an LMI problem in $\gamma, \kappa_Y, \kappa_L, Y$ and L [11].

Problem 1. *Minimize* $a_1\gamma + a_2\kappa_Y + a_3\kappa_L$ *subject to*

$$Y > 0, \tag{8}$$

$$\begin{bmatrix} AY + YA^{\mathrm{T}} + BL + L^{\mathrm{T}} B^{\mathrm{T}} & I & YH^{\mathrm{T}} \\ I & -I & 0, \\ HY & 0 & -\gamma I \end{bmatrix} < 0 \tag{9}$$

$$\gamma - 1/\bar{\alpha}^2 < 0 \tag{10}$$

and

$$\begin{bmatrix} -\kappa_L I & L^T \\ L & -I \end{bmatrix} < 0; \quad \begin{bmatrix} Y & I \\ I & \kappa_Y I \end{bmatrix} > 0. \tag{11}$$

Several comments need to be made regarding this design procedure.

Remark 1. The control design is formulated as a convex optimization problem, which ensures computational efficiency. The gain matrix is obtained directly as $K = LY^{-1}$, with no need for trial and error procedures.

Remark 2. The norm of the gain matrix is implicitly constrained by inequalities (11), which imply that $\|K\| \leqslant \sqrt{\kappa_L \kappa_Y}$. This is necessary in order to prevent unacceptably high gains that an unconstrained optimization may otherwise produce [11,12].

Remark 3. If the LMI optimization is feasible, the resulting gain matrix stabilizes the closed-loop system for *all* nonlinearities satisfying (5). Condition (10) additionally secures that α is greater than some desired value $\bar{\alpha}$.

Remark 4. The obtained controllers are linear, so their implementation is straightforward and cost effective.

Although the control law obtained by solving Problem 1 is robust, its centralized nature makes it unsuitable for large-scale applications. In order to develop a strategy that can be efficiently implemented in a multiprocessor environment, let us assume that (4) can be represented as an interconnection of N subsystems

$$\begin{aligned} \dot{x}_i &= A_{ii}x_i + \sum_{j\neq i} A_{ij}x_j + h_i(x) + B_i u_i \quad (i = 1,2,\ldots,N), \\ y_i &= C_i x_i. \end{aligned} \tag{12}$$

In (12), $x_i \in R^{n_i}$ is the state of the ith subsystem, while $u_i \in R^{m_i}$ and $y_i \in R^{q_i}$ represent its input and output vectors. We will further assume that the states of each subsystem are *locally available*, and that communication channels exist between preassigned pairs of processors. Denoting the set of subsystems that transmit their state information to subsystem i by T_i, we will look for a control law in the form

$$u_i = K_{ii}x_i + \sum_{j\in T_i} K_{ij}x_j \quad (i = 1,2,\ldots,N). \tag{13}$$

In order to adapt Problem 1 to this type of control scheme, it is necessary to introduce two additional requirements.

Requirement 1. Matrix L has the general structure

$$L = \begin{bmatrix} L_{11} & \cdots & L_{1N} \\ \vdots & \ddots & \vdots \\ L_{N1} & \cdots & L_{NN} \end{bmatrix} \tag{14}$$

with $L_{ii} \neq 0$, $\forall i$, and $L_{ij} \neq 0$ if and only if $j \in T_i$.

Requirement 2. Matrix $Y = \text{diag}\{Y_{11}, \ldots, Y_{NN}\}$ is block diagonal, with blocks Y_{ii} whose sizes are compatible with those of L_{ii}. When these modifications are incorporated into Problem 1, it is easily verified that the gain matrix consists of N diagonal blocks

$$K_{ii} = L_{ii}Y_{ii}^{-1} \tag{15}$$

and off-diagonal blocks defined as

$$K_{ij} = \begin{cases} L_{ij}Y_{jj}^{-1}, & j \in T_i, \\ 0, & j \notin T_i. \end{cases} \tag{16}$$

The implementation of such a control using N processors is straightforward, given that the communication channels are preassigned.

3. Decentralized control with a low-rank centralized correction

In order to extend the design method proposed in the previous section, let us once again consider model (12), this time with the assumption that all subsystems can exchange information with a *single* front end processor. For such a system, we propose to design a feedback of the form

$$u_i = K_{ii}x_i + W_i V x \quad (i = 1,2,\ldots,N), \tag{17}$$

where W_i and V are matrices of dimension $m_i \times r$ and $r \times n$, respectively (with $r \ll n$). Defining the $m \times r$ matrix $W = [W_1^T, \ldots, W_N^T]^T$, the overall control law can be expressed as

$$u = (K_D + WV)x, \tag{18}$$

where $K_D = \text{diag}\{K_{11}, \ldots, K_{NN}\}$ corresponds to decentralized feedback, and the product WV represents a *low-rank centralized correction.*

To obtain a control of the form (18), we will look for a solution of Problem 1 in the form

$$Y = Y_D + UY_CU^T, \quad L = L_DU^T, \tag{19}$$

where

(1) Y_D is an unknown symmetric block diagonal matrix, with blocks of dimension $n_i \times n_i$.
(2) L_D is an unknown block diagonal matrix, with blocks of dimension $m_i \times n_i$.
(3) U is a fixed $n \times r$ matrix.
(4) Y_C is an unknown symmetric $r \times r$ matrix.

For any given choice of U, Problem 1 thus becomes an LMI optimization in $\gamma, \kappa_Y, \kappa_L, Y_D, Y_C$ and L_D. To see the connection between (19) and the desired feedback structure, we should observe that Y^{-1} can be expressed using the Sherman–Morrison formula as (e.g. [14])

$$Y^{-1} = Y_D^{-1} - SRU^TY_D^{-1} \tag{20}$$

with

$$S = Y_D^{-1}UY_C, \quad R = [I + U^TS]^{-1}. \tag{21}$$

Since $K = LY^{-1}$, it is easily verified that this matrix can be factorized as $K = WV$, where

$$W = L_D(I - U^TSR) \tag{22}$$

and

$$V = U^TY_D^{-1} \tag{23}$$

are matrices of dimension $m \times r$ and $r \times n$, respectively.

The computational complexity of the modified LMI problem is similar to the decentralized case, since the only added variable is the $r \times r$ matrix Y_C. The implementation in a multiprocessor environment is also quite straightforward. Indeed, if matrices W and V are partitioned as

$$W = [W_1^T, \ldots, W_N^T]^T \quad V = [V_1, \ldots, V_N] \tag{24}$$

the corresponding control scheme for processor i has the form shown in Fig. 1.

In this scheme, processor i performs multiplications involving matrices W_i, V_i and K_{ii}, which are of dimension $m_i \times r, r \times n_i$ and $n_i \times n_i$, respectively.

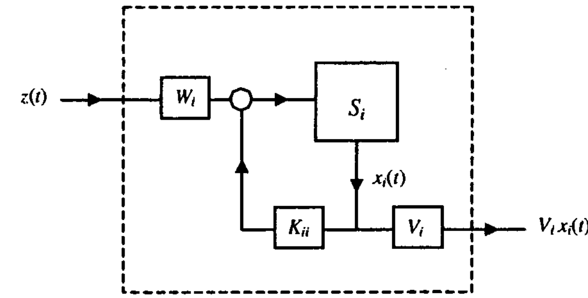

Fig. 1. Computation tasks for processor i.

Recalling that $r \ll n$, it follows that the required computational effort per processor is very modest. As for the front end processor, its main function is to assemble and distribute the subsystem information, and to form vector

$$z(t) = \sum_{j=1}^{N} V_jx_j(t). \tag{25}$$

The only communication tasks involved are single-node gather and scatter operations, which result in low overhead [15].

The following simple example illustrates how the control strategy proposed in (18) can improve the system performance.

Example 1. Let us consider a system of the form (12) with

$$A = \begin{bmatrix} 0 & -2 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0.5 & -2 & 0 & 0 \\ -0.1 & 1 & -2 & 0 & -0.5 & 0.9 \\ 0 & 1 & 0 & 0 & -0.5 & 0 \\ 0 & -0.1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0.8 & 0.2 & -1 & -1 \end{bmatrix}; \quad B_D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}. \tag{26}$$

We will assume that $h(x)$ is *uncertain*, and our objective will be to stabilize the system for *any* nonlinearity that satisfies inequality (5). In this process, we the robustness parameter α needs to be maximized.

Setting $H = I$, the decentralized control obtained by solving Problem 1 has the form $u = K_Dx$, with

$$K_D = \begin{bmatrix} -1.09 & -3.38 & 0.79 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4.31 & 0.32 & 1.14 \end{bmatrix}. \tag{27}$$

The corresponding robustness bound was found to be $\alpha = 0.263$, which implies that the closed-loop system is stable for *any* nonlinearity that satisfies

$$h^{\mathrm{T}}(x)h(x) \leqslant (0.263)^2 x^{\mathrm{T}}x. \tag{28}$$

If Problem 1 is now solved with the modifications proposed in this section (using $r = 2$), we obtain a control law of the form (18), where

$$K_{\mathrm{D}} = \begin{bmatrix} -4.63 & -0.81 & 3.35 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3.66 & 0.08 & 1.66 \end{bmatrix}, \tag{29}$$

$$W = \begin{bmatrix} -0.06 & 3.54 \\ 0.98 & -10.64 \end{bmatrix} \tag{30}$$

and

$$V = \begin{bmatrix} 9.37 & 0.89 & -10.63 & -1.03 & -0.04 & 0.89 \\ 0.94 & -0.43 & -1.05 & 0.25 & 0.08 & 0.19 \end{bmatrix}. \tag{31}$$

In this case, the robustness bound becomes $\alpha = 0.512$, which is nearly twice as large as in the decentralized case. This suggests that a low-rank centralized correction can significantly improve the system robustness with respect to uncertain nonlinearities.

4. BBD output control

As a further generalization of the LMI-based design, in this section we will consider the case when the multiprocesssor architecture is *unspecified*. The only assumption regarding matrix A is that it is sparse, which is realistic for most practical large-scale models. We will also assume that there are m inputs and q outputs, but their location is to be determined by the designer.

In order to develop an appropriate control scheme for such a system, let us first observe that any sparse matrix can be reordered into the bordered block diagonal (BBD) form shown in Fig. 2. Although such a structure is well suited for parallel processing (e.g. [16,17]), identifying an appropriate permutation matrix turns out to be a difficult graph-theoretic problem. Among the many heuristic schemes that have been developed for this purpose we single out the algorithm in [18], which can produce a prescribed number of balanced diagonal blocks. This method was found to be effective over a wide range of nonzero patterns, and for matrices as large as 500,000 × 500,000.

Given $N = \min(m, q)$, the first step of the proposed design process is to permute matrix A into a BBD form with N blocks of balanced size. The location of inputs and outputs will then be chosen in such a way that each diagonal block has *at least* one input and one output. This choice obviously determines

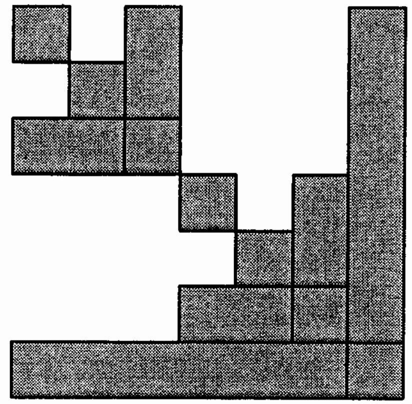

Fig. 2. Matrix A with a nested BBD structure.

the block diagonal input and output matrices $B_{\mathrm{D}} = \mathrm{diag}\{B_{11}, \ldots, B_{NN}\}$ and $C_{\mathrm{D}} = \mathrm{diag}\{C_{11}, \ldots, C_{NN}\}$. With such a partitioning, the system takes the form

$$\begin{aligned} \dot{x}_i &= A_{ii}x_i + A_{iN}x_N + h_i(x) + B_iu_i, \\ y_i &= C_ix_i \quad (i = 1, 2, \ldots, N-1) \end{aligned} \tag{32}$$

and

$$\begin{aligned} \dot{x}_N &= \sum_{j=1}^{N} A_{Nj}x_j + h_N(x) + B_Nu_N, \\ y_N &= C_Nx_N. \end{aligned} \tag{33}$$

In view of this structure, our objective in the following will be to obtain an *output BBD control law*

$$\begin{aligned} u_i &= K_{ii}y_i + K_{iN}y_N \quad (i = 1, 2, \ldots, N-1), \\ u_N &= \sum_{j=1}^{N} K_{Nj}y_j. \end{aligned} \tag{34}$$

We propose to accomplish this by modifying Problem 1 in such a way that the product LY^{-1} can be factorized as

$$LY^{-1} = KC, \tag{35}$$

where K is a BBD matrix.

In order to satisfy condition (35), we will look for a solution of Problem 1 in the form

$$Y = Y_0 + UY_{\mathrm{C}}U^{\mathrm{T}}, \quad L = L_{\mathrm{C}}U^{\mathrm{T}}, \tag{36}$$

where U is a fixed $n \times q$ matrix, Y_0 and Y_{C} are unknown symmetric matrices of dimensions $n \times n$ and $q \times q$, respectively, and L_{C} is an unknown $m \times q$ matrix. For any given choice of U, the optimization (8)–(11) thus becomes an LMI problem in $\gamma, \kappa_Y, \kappa_L, Y_0, Y_{\mathrm{C}}$ and L_{C}.

Observing that matrices B_{D} and C_{D} consist of $m_i \times n_i$ and $q_i \times n_i$ diagonal blocks, respectively, we need to add the following requirements to Problem 1:

Requirement 1. $U = \mathrm{diag}\{U_1, \ldots, U_N\}$ is a fixed, user-defined block diagonal matrix, with blocks U_i of dimension $n_i \times q_i$.

Requirement 2. $Y_0 = \mathrm{diag}\{Y_0^{(1)}, \ldots, Y_0^{(N)}\}$ and $Y_{\mathrm{C}} = \mathrm{diag}\{Y_{\mathrm{C}}^{(1)}, \ldots, Y_{\mathrm{C}}^{(N)}\}$ are unknown symmetric block diagonal matrices. The dimensions of $Y_0^{(i)}$ and $Y_{\mathrm{C}}^{(i)}$ are $n_i \times n_i$ and $q_i \times q_i$, respectively.

Requirement 3. L_{C} is an unknown BBD matrix, with the structure

$$L_{\mathrm{C}} = \begin{bmatrix} L_{11} & 0 & \ldots & L_{1N} \\ 0 & L_{22} & \ldots & L_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ L_{N1} & L_{N2} & \ldots & L_{NN} \end{bmatrix}. \tag{37}$$

In this matrix, block L_{ij} has dimension $m_i \times q_j$.

Requirement 4. Matrix Y_0 must satisfy

$$U^{\mathrm{T}}Y_0^{-1} = C_{\mathrm{D}} \tag{38}$$

which implies

$$Y_0 C_{\mathrm{D}}^{\mathrm{T}} = U. \tag{39}$$

This is an additional *equality constraint* that needs to be incorporated into the LMI optimization. The simplest way to accomplish this is to look for Y_0 in the form

$$Y_0 = Q_{\mathrm{D}} Y_{\mathrm{Q}} Q_{\mathrm{D}}^{\mathrm{T}} + C_{\mathrm{D}}^{\mathrm{T}}(C_{\mathrm{D}}C_{\mathrm{D}}^{\mathrm{T}})^{-1}C_{\mathrm{D}}, \tag{40}$$

where Q_{D} is an $n \times (n-q)$ block diagonal matrix such that

$$Q_{\mathrm{D}}^{\mathrm{T}} C_{\mathrm{D}}^{\mathrm{T}} = 0. \tag{41}$$

In that case, we need to compute an $(n-q) \times (n-q)$ matrix Y_{Q}, which is symmetric and block diagonal, with $(n_i - q_i) \times (n_i - q_i)$ blocks.

According to the Sherman–Morrison formula, from (36) and (38) it follows that

$$LY^{-1} = KC_{\mathrm{D}}, \tag{42}$$

where

$$K = L_{\mathrm{C}}(I - U^{\mathrm{T}}SR). \tag{43}$$

Furthermore, the structure of U, Y_{Q}, and Y_{C} implies that matrix $I - U^T SR$ must be block diagonal, with blocks of dimension $q_i \times q_i$. Consequently, given the BBD structure of matrix L_{C}, it follows that the corresponding gain matrix is guaranteed to have the same form.

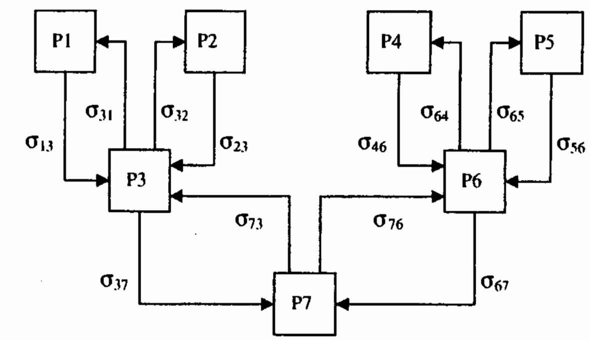

Fig. 3. The overall communication scheme.

Table 1
Description of the communication tasks

Task	Description
σ_{13}	Send $K_{31}y_1$ and $K_{71}y_1$ to P3
σ_{23}	Send $K_{32}y_2$ and $K_{72}y_2$ to P3
σ_{31}	Send $K_{13}y_3$ and $K_{17}y_7$ to P1
σ_{32}	Send $K_{23}y_3$ and $K_{27}y_7$ to P2
σ_{37}	Send $(K_{71}y_1 + K_{72}y_2 + K_{73}y_3)$ to P7
σ_{46}	Send $K_{64}y_4$ and $K_{74}y_4$ to P6
σ_{56}	Send $K_{65}y_5$ and $K_{75}y_5$ to P6
σ_{64}	Send $K_{46}y_6$ and $K_{47}y_7$ to P4
σ_{65}	Send $K_{56}y_6$ and $K_{57}y_7$ to P5
σ_{67}	Send $(K_{74}y_4 + K_{75}y_5 + K_{76}y_6)$ to P7
σ_{73}	Send $K_{17}y_7, K_{27}y_7$ and $K_{37}y_7$ to P3
σ_{76}	Send $K_{47}y_7, K_{57}y_7$ and $K_{67}y_7$ to P6

The resulting output BBD control (34) can be easily generalized to nested BBD structures. Such a control is inherently hierarchial, and can easily be embedded into a tree-type multiprocessor architecture. For the matrix in Fig. 2 (which is nested and has 7 diagonal blocks), the overall communication scheme would be the one shown in Fig. 3, with the communication tasks described in Table 1.

5. Conclusions

In this paper, we considered an LMI-based strategy for robust control design in large-scale systems. The proposed control laws are suitable for a multiprocessor environment, and can accommodate several generic types of information exchange, while minimizing the communication overhead. In all cases, the underlying LMI optimization requires only a modest computational effort.

Acknowledgement

The work reported in this paper was supported by NSF grant ECS-0099469.

References

[1] D.D. Šiljak, A.I. Zecevic, Large-scale and decentralized systems, in: J.G. Webster (Ed.), Wiley Encyclopedia of Electrical and Electronics Engineering, Wiley, New York, 1999.
[2] D.D. Šiljak, Decentralized Control of Complex Systems, Academic Press, Cambridge, 1991.
[3] N. Rouche, P. Habets, M. Laloy, Stability Theory by Lyapunov's Direct Method, Springer, New York, 1977.
[4] M.E. Sezer, D.D. Šiljak, Nested ε decompositions and clustering of complex systems, Automatica 22 (1986) 321–331.
[5] M.E. Sezer, D.D. Šiljak, Nested epsilon decompositions of linear systems: weakly coupled and overlapping blocks, SIAM Journal of Matrix Analysis and Applications 12 (1991) 521–533.
[6] A.I. Zečević, D.D. Šiljak, A block-parallel Newton method via overlapping epsilon decompositions, SIAM Journal of Matrix Analysis and Applications 15 (1994) 824–844.
[7] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory, SIAM, Philadelphia, PA, 1994.
[8] J.C. Geromel, J. Bernussou, P. Peres, Decentralized control through parameter space optimization, Automatica 30 (1994) 1565–1578.
[9] J.C. Geromel, J. Bernussou, M.C. de Oliveira, H_{infty}-norm optimization with constrained dynamic output feedback controllers: decentralized and reliable control, IEEE Transactions on Automatic Control 44 (1999) 1449–1454.
[10] L. El Ghaoui, S. Niculescu (Eds.), Advances in Linear Matrix Inequalities Methods in Control, SIAM, Philadelphia, PA, 2000.
[11] D.D. Šiljak, D.M. Stipanović, Robust stabilization of nonlinear systems: the LMI approach, Mathematical Problems in Engineering 6 (2000) 461–493.

[12] D.D. Šiljak, D.M. Stipanović, A.I. Zečević, Robust decentralized turbine/governor control using linear matrix inequalities, IEEE Transactions on Power Systems 17 (2002) 715–722.
[13] A.I. Zečević, D.D. Šiljak, Stabilization of nonlinear systems with moving equilibria, IEEE Transactions on Automatic Control 48 (2003) 1036–1040.
[14] G. Golub, C. van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, 1996.
[15] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation, Prentice-Hall, Englewood Cliffs, 1989.
[16] I.S. Duff, A.M. Erisman, J.K. Reid, Direct Methods for Sparse Matrices, Clarendon Press, Oxford, 1986.
[17] A. George, J.R. Gilbert, J.W.H. Liu (Eds.), Graph Theory and Sparse Matrix Computation, Springer-Verlag, New York, 1993.
[18] A.I. Zečević, D.D. Šiljak, "Balanced decompositions of sparse systems for multilevel parallel processing, IEEE Transactions on Circuits and Systems—Part I 41 (1994) 220–233.