Adv Syst Sci Appl 2018; **2**:11–25
Published online at http://ijassa.ipu.ru/index.php/ijassa/article/view/604

Surrogates for the matrix ℓ_0-quasinorm in sparse feedback design: Numerical study of the efficiency

Alexey Bykov[1], Pavel Shcherbakov[1,2]*

[1] *Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*
[2] *Institute for Systems Analysis, Federal Research Center "Computer Science and Control", Russian Academy of Sciences, Moscow, Russia*

Abstract: Some formulations of the optimal control problem require the resulting controller to be sparse; i.e., to contain zero elements in the gain matrix. On one hand, sparse feedback leads to the drop of performance as compared to the optimal control; on the other hand, it confers useful properties to the system. For instance, sparse controllers allow to design distributed systems with decentralized feedback. Some sparse formulations require the gain matrix of the controller to have a special sparse structure which is characterized by the presence of zero rows in the matrix. In this paper, various approximations to the number of nonzero rows of a matrix are considered and applied to sparse feedback design in optimal control problems for linear systems. Along with a popular approach based on using the matrix ℓ_1-norm, more complex nonconvex surrogates are proposed and discussed, those surrogates being minimized via special numerical procedures. The efficiency of the approximations is compared via numerical experiments.

Keywords: sparse control, ℓ_1-optimization, linear systems, optimal control, linear matrix inequalities

1. INTRODUCTION

Optimal control problems in linear theory may have a requirement imposed on the structure of the controller along with the necessity of optimizing the objective function. One example of such requirement would be *sparsity* of the controller, which is understood as having many zero elements in its gain matrix; e.g., see [12–14, 22]. Elementwise and block sparsity allows to design distributed systems with decentralized control. This additional sparsity requirement naturally leads to a drop in performance, which is explained by the fact that the set of sparse controllers is contained in the feasible set for the corresponding classical formulation. Nevertheless, in some formulations of control problems, a reasonable drop in performance is allowed for the sake of having a sparse control. For instance, the objective function in the standard LQR problem has the form of a quadratic functional, which is related to energy consumption in the system; e.g. fuel consumption of an aircraft. On one hand, a sparse control leads to worse (higher) values of the objective function, i.e., fuel consumption is increased; on the other hand, it augments the system with various useful properties. For example, decentralized control makes the system more reliable and fail-safe; decreasing the number of active actuators allows to slow down an equipment deterioration, thereby increasing the lifetime of the system.

In [15], a special kind of sparsity was introduced, which is different from a typical elementwise sparsity. The authors considered a controller to be sparse if its gain matrix has

*Corresponding author: alexey.bykov.mipt@gmail.com, cavour118@mail.ru

many zero rows or columns. Such controllers facilitate ease of hardware implementation of the control systems, such as a reduction of the number of actuators or sensors and the amount of information transmitted via control channels. A direct minimization of the number of nonzero rows or columns of a matrix is known to be NP-hard, since it involves combinatorial search. As the dimensions of the problem increase, the direct approach becomes inapplicable due to its exponential time complexity. In [15, 16], an approach was proposed which is based on solving convex *surrogate* problem instead of the original nonconvex one. This substitution is implemented using special matrix norms which are convex approximations (also called surrogates) to the number of nonzero rows/columns of a matrix (nonconvex matrix ℓ_0-quasinorms). This heuristic, however, does not guarantee an occurrence of zero rows/columns in the gain matrix, and by all appearances strict results cannot be obtained due to nonconvexity of the original problem. Nevertheless, the exploited heuristic is shown to be efficient and the approach seems to work for numerous examples. Such non-strict approaches require computer simulations for proving their efficiency, and in the current paper we put emphasis on the numerical study.

Apart from convex surrogates, some papers [2–4, 13] suggest using nonconvex approximations for more efficient detection of sparse solutions in case of vector and matrix variables.

The main goal of this paper consists in comparing the efficiency of different approximations to the matrix ℓ_0-quasinorm, which can be used for sparse feedback design in optimal control problems. Also, a general scheme for gaining trade-off between optimality and sparsity of the solution is proposed which allows to use various approximations in a similar way.

The key aspects of this paper are the formulation of several surrogates for the matrix ℓ_0-quasinorm, which can be used for promoting matrix row-sparsity, designing the scheme of numerical study, and the analysis of the results of experiments. Both models of simple systems and linearized models of real systems were chosen as test cases for conducting the numerical experiment. Also, various software for numerical computing [5, 18, 19], and algorithms for nonconvex optimization [3, 21] were used.

Numerical study is the central part of this paper due to the following reason. Although the approach based on convex surrogates heuristic is said to be efficient, one cannot be sure about getting zero rows in the gain matrix. Thus, different surrogates demonstrate different efficiency for different problems; this paper aims to study the efficiency of various approximations.

Our paper essentially exploits Linear Matrix Inequality (LMI, [1]) technique, which allows to formulate many classical problems from control theory in the form of SemiDefinite Programs (SDP, [20]).

2. SPARSE FEEDBACK DESIGN IN OPTIMAL CONTROL PROBLEMS

Consider the following control system:

$$\dot{x} = Ax + Bu, \; x(0) = x_0, \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^p$ is the control input, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, and the pair (A, B) is assumed to be controllable.

If system (2.1) is not stable, a stabilizing control is to be found, which also has to minimize an objective function along the system's trajectories. Various formulations differ in the form of the control input; in this paper we exploit a static state linear feedback form:

$$u = Kx, \; K \in \mathbb{R}^{p\times n}. \tag{2.2}$$

From this point, we consider the LQR (Linear Control Regulation) problem, the approach and the numerical experiment being applicable to other optimal control problems. In the LQR problem, the objective function has the form of the following quadratic functional:

$$J = \int_0^\infty \left(x^\top Rx + u^\top Su\right) dt, \tag{2.3}$$

where $R \in \mathbb{R}^{n\times n}$ and $S \in \mathbb{R}^{p\times p}$ are positive–definite matrices.

It is worth emphasizing that minimizing functional (2.3) automatically leads to stabilizing system (2.1). Indeed, if the system was unstable, integral (2.3) would be divergent; however, since the pair (A, B) is controllable, there exists a controller, which makes the functional J finitely valued.

The standard way of solving the LQR problem consists in finding a solution to the associated algebraic Riccati equation; however, in this paper we adhere to another approach (e.g., see [7]) based on the linear matrix inequality technique. In [7], the classical LQR problem is reduced to the following SDP problem:

Problem 1:
Let P_{opt}, Y_{opt} be the solution of the SDP

$$\operatorname{tr} P \longrightarrow \max \quad \textit{s.t.}$$
$$\begin{pmatrix} AP + PA^\top + BY + Y^\top B^\top & P & Y^\top \\ * & -R^{-1} & 0 \\ * & * & -S^{-1} \end{pmatrix} \preccurlyeq 0, \tag{2.4}$$

then the controller (2.2) *with gain matrix*

$$K_{opt} = Y_{opt} P_{opt}^{-1} \tag{2.5}$$

stabilizes system (2.1)*, the objective functional* (2.3) *being minimized, and the optimal value of the functional is equal to*

$$J_{opt} = x_0^\top P_{opt}^{-1} x_0. \tag{2.6}$$

This approach involving LMIs can be very useful, for instance, for robust formulations where the matrices of the system contain uncertainty. But what's more important for the current paper, it is the LMI-based approach that allows to solve the problem in sparse formulations.

As it was mentioned, the sparse formulation that we consider, imposes an additional requirement on the structure of the gain matrix; namely the intention is to have as many zero rows in matrix K as possible. Clearly, due to the structural constraint on the matrix K, the feasible set of controllers becomes smaller, so that a drop in performance of the sparse controller is unavoidable, and we do not want it to be significant.

The problem of comparison of the two controllers with respect to a quadratic performance index is not as straightforward as it may seem; e.g., see [8]. If one compares the values of the functional $J(P) = x^\top P^{-1}x$ corresponding to the two different stabilizing controllers K_1 and K_2 with the corresponding matrices P_1 and P_2, the sign of the inequality $J(P_1) \lesseqgtr J(P_2)$ may change depending on the initial condition x_0. In this paper we compare sparse controllers with the optimal ones, which, by construction, yield the optimal value of the functional J for *any* initial condition x_0. In other words, the sparse controller with matrix P_{sp} will always lose the optimal controller with matrix P_{opt} in terms of the value of the quadratic functional J. The magnitude of the loss (performance drop) for a given initial condition x_0 is

$$\frac{J_{sp}}{J_{opt}} = \frac{x_0^\top P_{sp}^{-1} x_0}{x_0^\top P_{opt}^{-1} x_0};$$

this value, obviously, can change significantly for various x_0.

There is a well-known approach [11] based on averaging the values of J over initial conditions uniformly distributed on the unit n-dimensional sphere. It is not hard to show that the mathematical expectation of J is equal to

$$\mathsf{E}(x_0^\top P^{-1} x_0) = \frac{1}{n}\operatorname{tr} P^{-1};$$

thus the magnitude of the sparse-to-optimal control loss "on the average" equals

$$\frac{J_{sp}}{J_{opt}} = \frac{\operatorname{tr} P_{sp}^{-1}}{\operatorname{tr} P_{opt}^{-1}}. \tag{2.7}$$

In this paper we adhere to the described approach based on "averaging"; i.e., we use relation (2.7).

In [16], a procedure is proposed to design sparse controllers in optimal control problems, in particular, in the LQR problem. The key stage of the procedure, which is responsible for the detection of the sparse structure of the controller, consists in minimizing the 1∞-norm of the gain matrix. This norm plays the role of a convex surrogate for the number of nonzero rows of a matrix and is defined as follows:

$$\|X\|_{1\infty} = \sum_{i=1}^{n} \max_{1\leqslant j\leqslant p} |x_{ij}|, \;\; X \in \mathbb{R}^{n\times p}. \tag{2.8}$$

The three-step procedure from [16], generalized to the case of arbitrary approximation to the matrix ℓ_0-quasinorm, was used as a basis for the numerical study in this paper. The key aspects of the procedure are given below.

Step 1 (optimal control design). The classical LQR problem is solved (**Problem 1**) and the value $\operatorname{tr} P_{opt}^{-1}$ is fixed. This value corresponds to the optimal value J_{opt} of the quadratic functional J.

Step 2 (detection of sparse structure). Let $\alpha > 1$ denote the maximum admissible loss in the quadratic performance index that appears due to sparsity; i.e., the following condition is imposed:

$$\frac{\operatorname{tr} P_{sp}^{-1}}{\operatorname{tr} P_{opt}^{-1}} \leqslant \alpha, \;\; \alpha > 1.$$

So, α is the coefficient of the allowed performance drop "on average" and it acts as the parameter of the following SDP problem:

Problem 2:

$$\begin{gathered} \|Y\|_{(sp)} \longrightarrow \min \quad \textit{s.t.} \quad (2.4), \\ \begin{pmatrix} X & I \\ I & P \end{pmatrix} \succcurlyeq 0, \; \textit{and} \; \operatorname{tr} X \leqslant \alpha \operatorname{tr} P_{opt}^{-1}. \end{gathered} \tag{2.9}$$

Note that, similarly to **Problem 1**, the gain matrix K does not explicitly appear in the constraints or in the objective function of the SDP. Thus, the variables are positive–definite matrices P and X and an auxiliary matrix variable $Y = KP$; the latter was introduced to transform the constraints to the form of *linear* matrix inequalities.

The minimized function in **Problem 2** is denoted by $\|\cdot\|_{(sp)}$ and it plays the role of an approximation to the matrix ℓ_0-quasinorm. The 1∞-norm is one of such approximations and it allows to solve convex problem instead of the nonconvex one. There are more complex and efficient surrogates, some of which are described in the next section.

Let the matrix Y_0 be the solution of **Problem 2** and assume it contains rows with all elements being equal or close to zero. These rows are memorized and kept as strict zeros, which automatically means that the gain matrix $K_0 = Y_0 P^{-1}$ will also contain the same zero rows. The latter fact follows from the rules of the matrix multiplication. Note that the value of the objective function does not exceed αJ_{opt} "on average".

Step 3 (optimization over sparse-structured controllers). The LQR problem is again solved but with sparse structure of the controller being fixed as found at Step 2. In other words, **Problem 1** is solved with additional constraints on the matrix variable Y.

To sum up, the procedure described above gives a way to design a controller which may have a sparse structure at the expense of a reasonable drop in performance; thus, a trade-off between optimality and sparsity of the control can be achieved.

3. APPROXIMATIONS TO THE MATRIX ℓ_0-QUASINORM

So, the key step in the sparse feedback design procedure is the sparsity detection step. The detection is performed via minimizing a surrogate for the number of nonzero rows of a matrix.

Below we consider four various approximations to the matrix ℓ_0-quasinorm, efficiency of which is compared via numerical experiments. All approximations are described as functions of r_i, where r_i qualifies the magnitude of the values of the ith row of the matrix Y. If a row of the matrix is treated as a vector, then r_i can be, e.g., one of the standard vector norms: l_1-, l_2-, or l_∞-norm. A matrix row is called a *zero row* if and only if all of its elements are zeros. Also, if any of the mentioned norm approaches zero, then all of the row elements approach zero. Numerical experiments show that the choice of the particular norm is not essential, so, to be specific, we chose l_∞-norm, the maximum of the absolute values of the components. Thus, we define r_i as follows:

$$r_i = \max_{1 \leqslant j \leqslant n} |y_{ij}|, \quad Y = \|y_{ij}\| \in \mathbb{R}^{p \times n}. \tag{3.10}$$

3.1. 1∞*-norm*

The 1∞-norm (2.8) was introduced in [16]; in terms of r_i (3.10) it can be written as:

$$\|Y\|_{1\infty} = \sum_{i=1}^{p} r_i. \tag{3.11}$$

This function is convex, therefore its minimization subject to constraints (2.4), (2.9) is an SDP problem, which can be efficiently solved by means of convex optimization tools [20].

3.2. Weighted 1∞*-norm*

In [3], the authors developed the ideas of ℓ_1-optimization for promoting sparsity in case of vector variables. They introduced a new iterative procedure which consists in solving a sequence of *weighted* ℓ_1-optimization problems. At each iteration, the weight coefficients are updated due to the solution received at the previous step. The weighted 1∞-norm is defined as follows:

$$\|Y\|_{w_{1\infty}} = \sum_{i=1}^{p} w_i r_i. \tag{3.12}$$

The iterative procedure from [3] can be schematically written down as follows:

Algorithm 1:

1. *Initial estimate:* $l = 0,\ w_i^{(0)} = 1,\ i = 1, \dots, p.$
2. *The convex problem with currently accepted weights is solved:*

$$r^{(l)} = \arg\min_r \left[\sum_{i=1}^{p} w_i^{(l)} r_i \right] \quad \textit{s.t.} \quad (2.4), (2.9).$$

3. *The weights are updated:*

$$w_i^{(l+1)} = \frac{1}{|r_i^{(l)}| + \varepsilon}, \quad \varepsilon > 0.$$

4. *Quit the loop if the procedure converged or the maximum number of iterations is exceeded. If both conditions are false, go back to step 2.*

The authors of the original paper [3] confirmed via numerous examples that the new approach involving the minimization of the weighted ℓ_1-norm performs much better than the "standard" non-weighted approach. This can be explained by the fact that the sequence of convex problems can better approximate the original nonconvex problem of minimizing the matrix ℓ_0-quasinorm.

3.3. *Nonconvex sparsity detector,* NSD

In [2], a nonconvex sparsity-promoting surrogate was introduced:

$$\|Y\|_{\mathsf{NSD}} = \sum_{i=1}^{p} r_i \prod_{j \neq i} \frac{r_j}{r_j + 1}. \tag{3.13}$$

The authors provide some motivation for this particular approximation and demonstrate its efficiency as compared to the 1∞-norm.

The function (3.13) is shown to be a DC-function (Difference of Convex, see [6]) and for its minimization, the Concave-Convex Procedure (*abbr.* CCCP, e.g., see [21]) is proposed.

Let the function (3.13) be represented as a DC-function: $\mathsf{NSD}(r) = U(r) - V(r)$. Then the procedure for its minimization is given below:

Algorithm 2:

1. *Initial estimate:* $l = 0,\ r_i^{(0)} = \max_{1 \leqslant j \leqslant n} |y_{ij}|,\ i = 1, \dots, p,$ *where* $Y_{opt} = \|y_{ij}\|$ *is the solution of* ***Problem 1****.*
2. *The following convex problem is solved:*

$$r^{(l+1)} = \arg\min_r \left[U(r) - r^\top \nabla V\left(r^{(l)}\right) \right] \quad \textit{s.t.} \quad (2.4), (2.9).$$

3. *Quit the loop if the procedure converged or the maximum number of iterations is exceeded. If both conditions are false, go back to step 2.*

In [9], the results on the convergence of the CCCP procedure are presented; in the general case, global convergence is not guaranteed, since the problem is not convex. However, for many applications this procedure yields an appropriate solution.

3.4. *Nonconvex approximation* log-sum

In [3, 4, 13], the function log-sum (see definition below) is shown to present a more precise approximation to the ℓ_0-norm of a vector than convex approximations. Indeed, let us consider the following scalar functions (see Fig. 3.1) which can be used as penalty functions for nonzero values:

$$f_0(x) = 1_{[x\neq 0]}, \qquad f_1(x) = |x|, \qquad f_{\log,\varepsilon}(x) = C_\varepsilon \log\Big(1 + \frac{|x|}{\varepsilon}\Big);$$

the factor C_ε is chosen in a such way that the equality $f_{\log,\varepsilon}(1) = 1 = f_0(1) = f_1(1)$ holds.

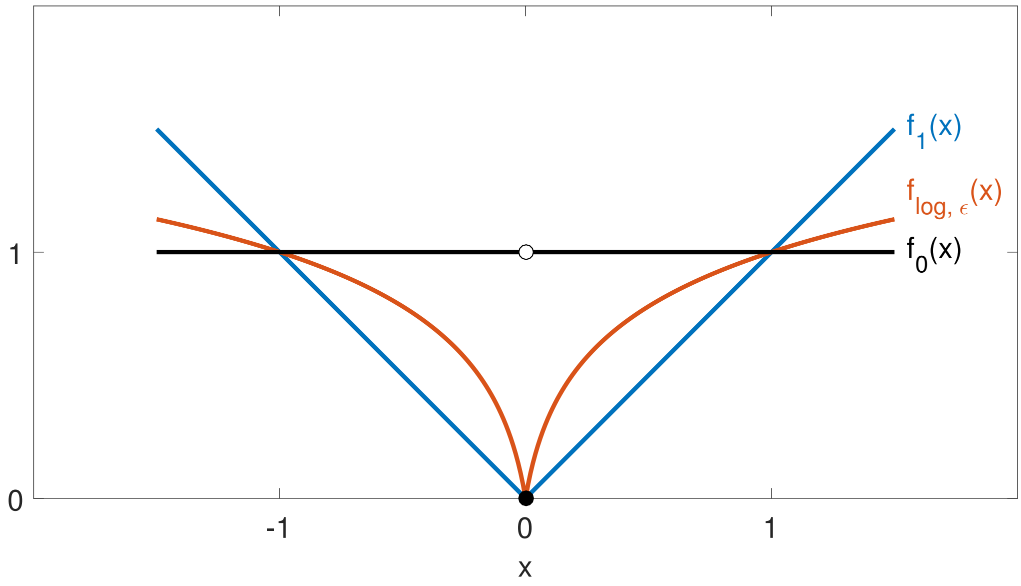

Fig. 3.1. Approximations to the ℓ_0-quasinorm

The derivative of the function $f_{\log,\varepsilon}(x)$ in the positive vicinity of the origin grows approximately like $\frac{1}{\varepsilon}$ as $\varepsilon \longrightarrow 0$. This leads to a relatively big amount of penalty for small but not exactly zero values. Moreover, note that $f_{\log,\varepsilon}(x) \longrightarrow f_0(x)$ as $\varepsilon \longrightarrow 0$. Unfortunately, as ε tends to zero, the minimization of the function log-sum becomes difficult; for heuristic of choosing ε see [3].

The function log-sum adapted for promoting matrix row-sparsity can be written as

$$\|Y\|_{\text{log-sum}} = \sum_{i=1}^{p} \log\left(1 + r_i\right). \tag{3.14}$$

Since the function (3.14) is concave, it cannot be minimized via convex optimization framework. However, it can be treated as a special case of DC-function, so, it can be minimized via concave-convex procedure (CCCP) just like the NSD function.

4. RESULTS OF NUMERICAL EXPERIMENTS

In the previous section, several well-known and relatively new approximations to the number of nonzero rows of a matrix were considered. Among these surrogates are the 1∞-norm exploited in [16], and the NSD function introduced in [2]. Both approximations, however, have not yet received enough approval from numerical experiments; therefore, they were chosen as contestants for the experiment in this paper. Also, two other approximations (the weighted 1∞-norm and the function $\log\text{-sum}$), which were earlier applied in the case of vector variables, were adapted for promoting matrix row-sparsity. All these surrogates were used at Step 2 of the three-step procedure to design sparse controllers in the numerical experiment.

All the examples described below were solved with MATLAB using the cvx framework [5] for solving convex optimization problems and the SDPT3 solver [19]. It was checked that using other solvers such as SeDuMi [18] did not give any specific benefits or drawbacks.

The main goal of the experiment was to compare the efficiency of approximations (3.11), (3.12), (3.13), (3.14) used for getting zero rows in the matrix Y_0 at Step 2 of the three-step procedure, which allows to design sparse controllers in the LQR problem. During the experiments, both models of simple systems and linearized models of real systems were considered. The latter were taken from the COMP$l_e ib$, a popular collection of test problems from control system design and related fields.

The general scheme of the experiment is as follows:

Algorithm 3 (Scheme of experiment)**:**

1. *The classical LQR problem is solved (**Problem 1**) and the value* $\mathrm{tr}\, P_{opt}^{-1}$ *is fixed. This value corresponds to the optimal value* J_{opt} *of the quadratic functional* J.
2. *The initial value of the maximum admissible loss in the quadratic performance index is chosen:* $\alpha_{min} = 1 + \varepsilon,\ \varepsilon > 0$.
3. *The magnitude of* α *is increased inside the loop until the most sparse control can be acquired. For every value of* α *the following steps are executed:*
   (a) *The detection of zero rows is performed (**Problem 2**) for a given* $\hat{\alpha}$. *The zero structure obtained from the solution is fixed; the number of zero rows denoted by* N_{nz} *is memorized.*
   (b) *The LQR problem (**Problem 1**) is solved with the sparse structure of the controller being fixed as found at the previous step. The performance drop is determined by the relation* $\alpha_{sp} = \frac{\mathrm{tr}\, P_{sp}^{-1}}{\mathrm{tr}\, P_{opt}^{-1}}$.

   Hence, every iteration of the loop gives us two points at the plane with coordinates the number of nonzero rows *and* the performance drop. *These points are* $(N_{nz}, \hat{\alpha})$ *and* (N_{nz}, α_{sp}), *where* $\hat{\alpha}$ *is the value of the maximum admissible loss which leads to the detection of zero rows, and* α_{sp} *is the actual loss in performance, which corresponds to the detected sparse structure of the controller. It is worth noting that* α_{sp} *might be drastically less than* $\hat{\alpha}$.
4. *The brute-force enumeration of all possible combinations of zero rows in the gain matrix is performed and for every combination and the corresponding sparse structure the LQR problem (**Problem 1**) is solved. Note that such a brute-force search is performed exceptionally within the experimental setup for illustration purposes only; it is not needed in the respective theorems. For the combination with index* $k \in [1; 2^p - 2]$ *the performance drop is determined by the relation* $\alpha_{bf,k} = \frac{\mathrm{tr}\, P_{bf,k}^{-1}}{\mathrm{tr}\, P_{opt}^{-1}}$.

Note that we consider $2^p - 2$ combinations of zero rows instead of 2^p, since two cases are excluded. The first one corresponds to the absence of zero rows, i.e., to the classical formulation without the sparsity requirement. The second obvious case corresponds to the situation where all rows of the gain matrix are zero rows.

The points (N_{nz}, α_{sp}) obtained as an output of the **Algorithm 3** are related to the Pareto optimality, a well-known concept from the theory of multiobjective optimization. Namely, a point is said to be Pareto-optimal if none of the criteria can be improved without worsening the others. Also, note that the values of α_{sp} might not be optimal for a given number of nonzero rows N_{nz}, since we solve the surrogate for the original nonconvex problem and the global convergence is not guaranteed. Nevertheless, varying the importance of one or another criterion we can gain a necessary trade-off between optimality and sparsity.

4.1. Examples from COMP$l_e ib$

The experiment was conducted in accordance with the described scheme for several problems from the COMP$l_e ib$ collection: "AC1", "AC9", "AC12", "HE3", "HE4" (linearized models of aircrafts and helicopters). These examples are of interest in terms of the dimension of the control input. The behaviour of various approximations to the matrix l_0-quasinorm for all five models is quite similar, so it is reasonable to present the results for just one example – model "HE4".

This eight-order model represents a twin-engine, multi-purpose military helicopter. Let us recall that the behaviour of the system is described by equations (2.1). The values of the entries of the matrices A and B are not presented for space considerations; they can be found in the COMP$l_e ib$ documentation. The weight matrices R and S in the quadratic functional (2.3) were set to identity.

The results of the experiment conducted in accordance with **Algorithm 3** are presented in Figs. 4.2, 4.3, and 4.4.

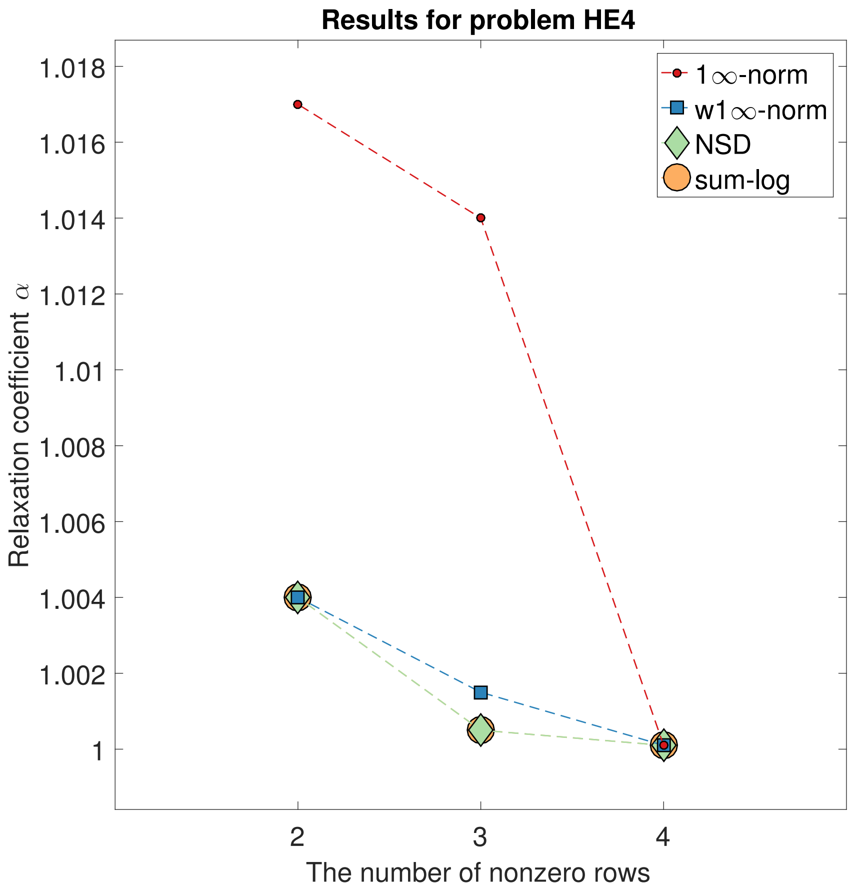

Fig. 4.2. "HE4". The detection of zero rows for various $\hat{\alpha}$

Figure 4.2 shows the points $(N_{nz}, \hat{\alpha})$ obtained for the functions (3.11), (3.12), (3.13), (3.14) in accordance with **Algorithm 3**. This plot demonstrates how big one should set the admissible loss in performance in order to get zero rows in the gain matrix of the controller. One can see that using approximations (3.12), (3.13), (3.14) leads to the detection of sparse structure for smaller values of $\hat{\alpha}$. It turned out that for any surrogate except for the 1∞-norm, in order to detect two zero rows out of the total four, we needed to set $\hat{\alpha}$ slightly less than 1.005, which corresponds to the performance drop of just 0.5%.

The transition from one zero row to two zero rows happens at close values of $\hat{\alpha}$, therefore using an insufficiently dense grid of $\hat{\alpha}$ values can mislead into believing that both zero rows appear simultaneously. Such simultaneous appearances of two zero rows were not noticed during the experiment, and they obviously should not happen due to the following reason. Every next zero row of the gain matrix makes the feasible set of controllers smaller, so the optimal value of the quadratic functional for the controller with $n + 1$ zero rows is a priori worse than the corresponding value for the controller with n zero rows.

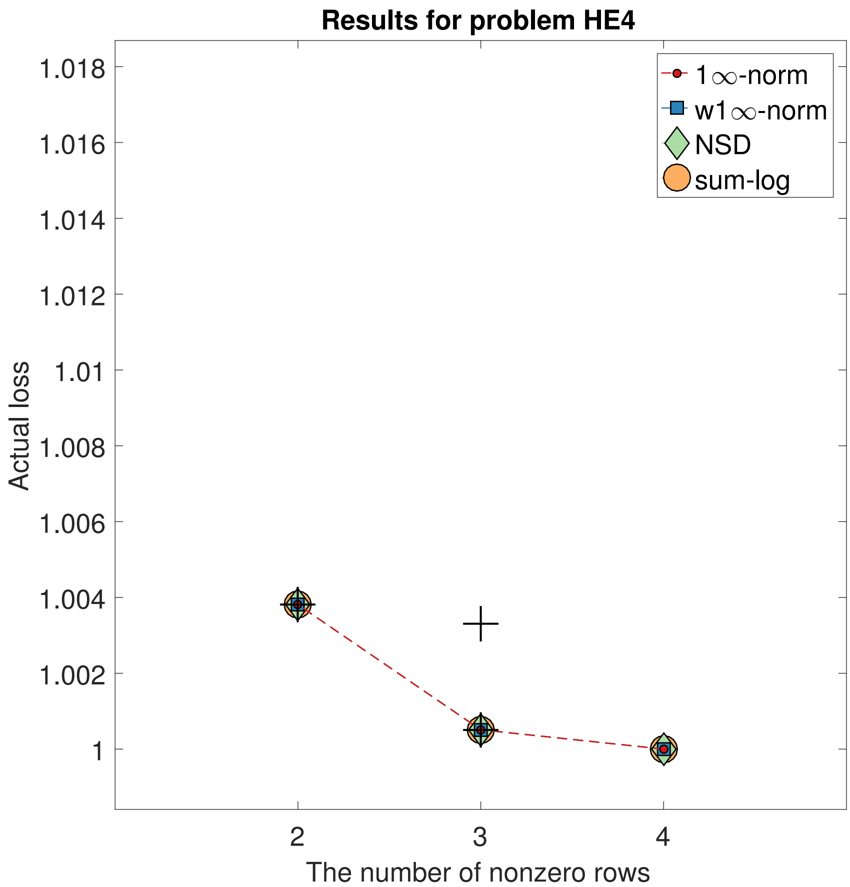

Fig. 4.3. “HE4”. The actual loss in performance index

Figure 4.3 shows the points (N_{nz}, α_{sp}) obtained from the **Algorithm 3**, which correspond to the actual sparse-to-optimal loss in performance. For the 1∞-norm we had $\hat{\alpha} \gg \alpha_{sp}$, while using the other surrogates yields a sparse structure for $\hat{\alpha}$ close to α_{sp}, which is surely an advantage of these approximations. Indeed, if one had to set $\hat{\alpha} \gg \alpha_{sp}$ in order to get a sparse control, the interpretation of $\hat{\alpha}$ as a maximum admissible performance loss would be incorrect. For instance, using the 1∞-norm in example “HE4”, in order to detect sparse control which yields 0.5% of the actual performance drop, we had to allow for a maximum admissible performance loss of $15 - 20\%$ during the detection step. Moreover, using the

1∞-norm in some examples can lead to a large difference between $\hat{\alpha}$ and α_{sp}. Using other approximations usually allows to detect a sparse structure for the values $\hat{\alpha}$ which are close to the corresponding values of α_{sp}; i.e., $\hat{\alpha}$ is a more accurate estimate of α_{sp}.

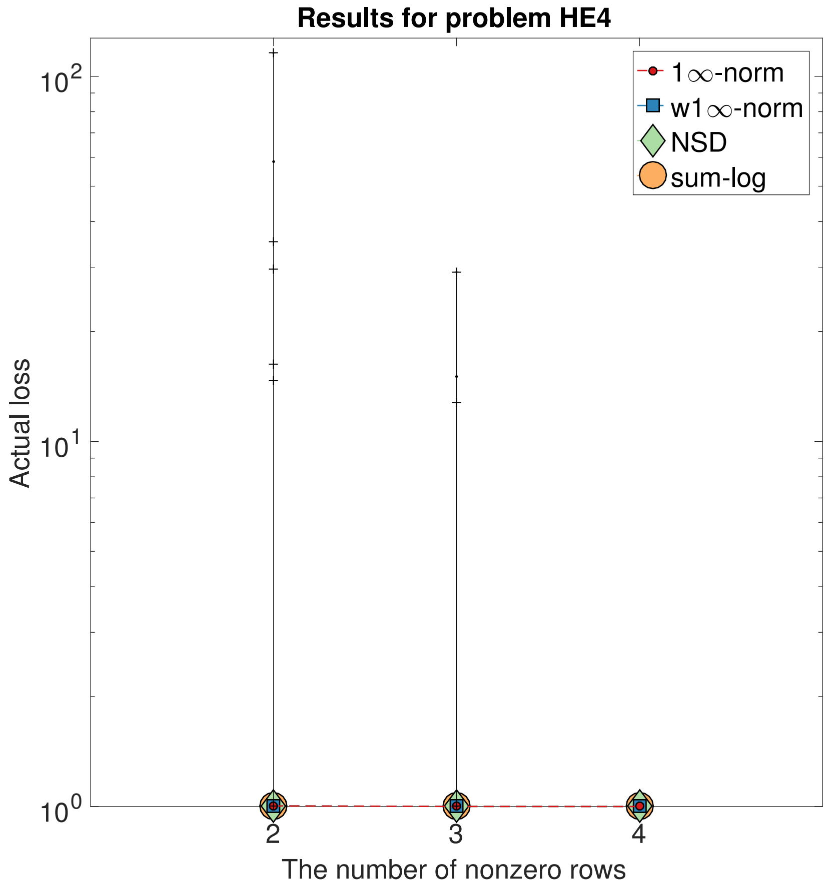

Fig. 4.4. "HE4". The actual loss in performance index. All possible combinations of zero rows

Figure 4.3 depicts the points marked as black crosses which were obtained at step 4 of the **Algorithm 3** during the brute-force enumeration. It turned out that for our example the three-step procedure yielded the optimal sparse structures for every number of zero rows. Figure 4.3 is just a part of the whole picture presented in Fig. 4.4, where all possible combinations of zero rows of the gain matrix can be seen. As we can see, a poor choice of the sparse structure can lead to a dramatic loss in the performance index. It is important to understand that the occurrence of large values of performance drop in Fig. 4.4 is explained by the specific properties of the system itself rather than by sparsity detection methods; some systems just do not tolerate zeroing out specific control inputs. This fact also leads to an interesting "side-effect" which consists in the ability to filter out sparse controllers with "poor" structure a priori by choosing appropriate values for $\hat{\alpha}$.

Similar results were obtained for other examples from COMP$l_e ib$ such as "AC1", "AC9", "AC12", "HE3". Therefore, all the observations and conclusions made for model "HE4" are also valid for all listed examples.

4.2. Mass Spring System

Consider the system which consists of N masses m_is connected via springs with stiffness coefficients k_is; the masses can slide frictionless along the line (see Fig. 4.5). Let us denote the displacement of the ith mass from its reference position as p_i and let the state variables be $x_1 = [p_1 \dots p_N]^\top$ and $x_2 = \dot{x}_1$.

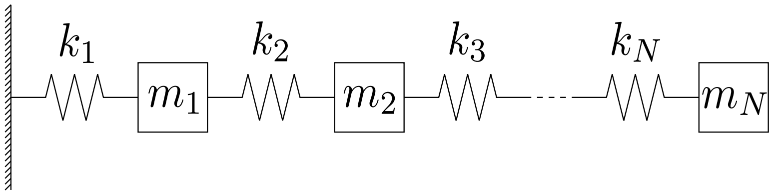

Fig. 4.5. Model "Mass Spring System"

Similarly to the previous section, the behaviour of the system is described by the equation (2.1). In this example we used identity weight matrices R and S in the quadratic functional (2.3).

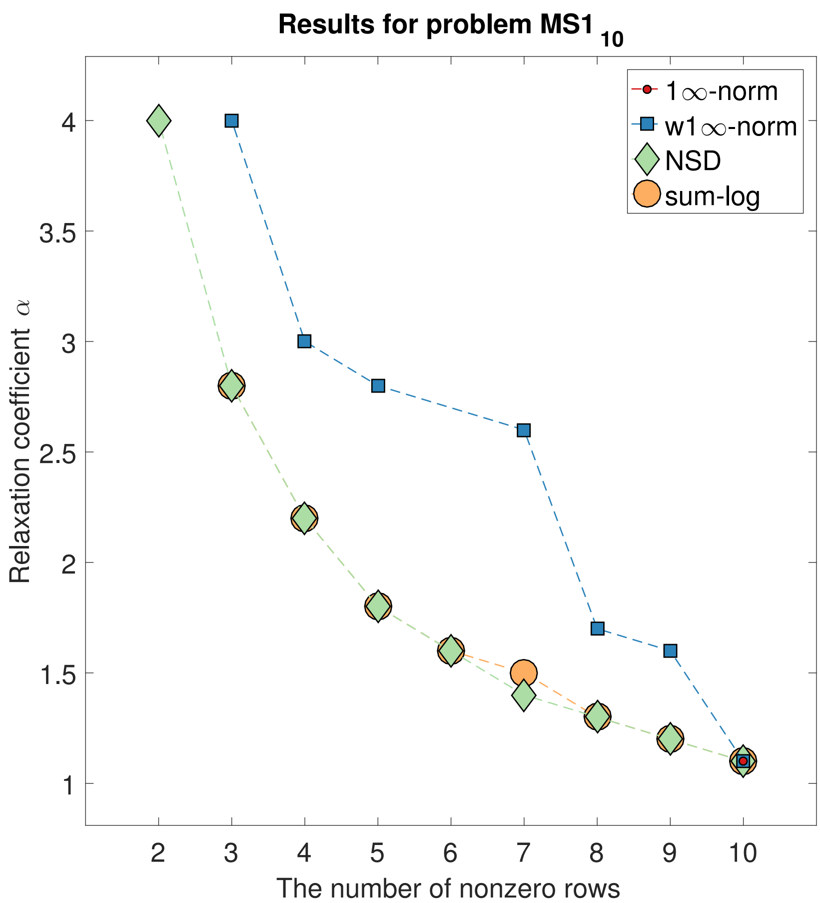

Fig. 4.6. "MS". The detection of zero rows for various $\hat{\alpha}$

For the sake of simplicity we considered the system with the following parameters:

$$m_1 = \dots = m_N = 1, \quad k_1 = \dots = k_N = 1,$$

so that the matrices of system (2.1) has the form

$$A = \begin{bmatrix} O & I \\ T & O \end{bmatrix}, \qquad B = \begin{bmatrix} O \\ I \end{bmatrix},$$

where $T \in \mathbb{R}^{N\times N}$ is a tridiagonal Toeplitz matrix with -2 on its main diagonal and 1 on its first super- and sub-diagonal, $I \in \mathbb{R}^{N\times N}$ is the identity matrix, and $O \in \mathbb{R}^{N\times N}$ is the zero matrix.

This model is useful, since it allows to choose arbitrary values for the number of masses N, i.e., to vary the dimension of the problem. The experiment was conducted for several values of N, the observed results being quite similar. We present the results for $N = 10$, since even for $N = 10$ the number of all possible combinations of zero rows is equal to $2^{10} - 2 = 1022$, thus, the brute-force approach is still possible yet challenging.

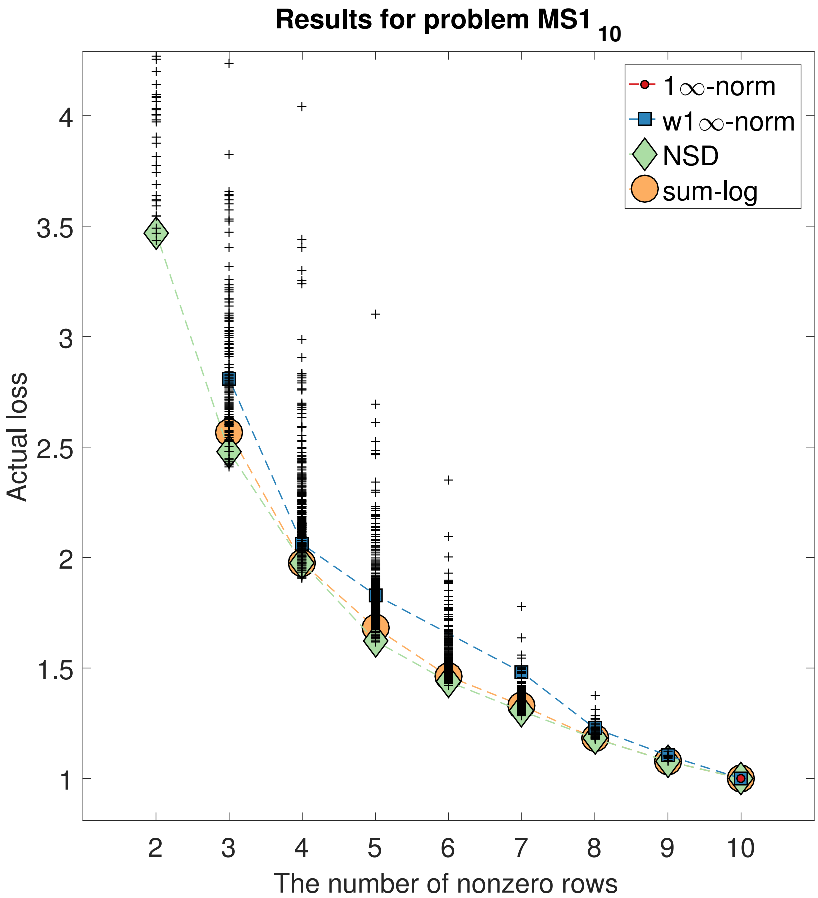

Fig. 4.7. "MS". The actual loss in performance index

The results of the experiment for this example are presented in Figs. 4.6 and 4.7. The figures show that use of the 1∞-norm in this problem did not lead to the successful detection of a sparse structure even for large values of $\hat{\alpha}$. This reminds us of the example "HE4" from the previous section, where we noticed that using the 1∞-norm makes $\hat{\alpha}$ a poor estimate of α_{sp}. This observation is valid for all values of N tested during the experiment. It is hard to explain such an inefficiency of the 1∞-norm. In this particular example one of the possible reasons might be a homogeneous internal structure of the system, which probably makes the choice of the control inputs to be zeroed out a challenging task.

Although the weighted 1∞-norm succeeded in yielding sparse controllers, it is less efficient than the nonconvex approximations (3.13) and (3.14). Nevertheless, **Algorithm 1** for minimizing the weighted 1∞-norm is quite straightforward in implementation and allows to achieve better results than the standard minimization of the 1∞-norm. There is an interesting detail about the process of the detection of zero rows. This process is sequential, i.e., the number of zero rows grows step-wise or one by one. If the values in the row approach zero, say become less than 10^{-10}, then, according to the algorithm, the weight for this row becomes very large, since it is inversely proportional to the maximum among the absolute values of the elements of this row. For a row with the weight 10^{10} it is hardly possible to gain nonzero values. Thus, if at any iteration some row becomes zero, it will stay zero till the end of the algorithm.

Implementation of the algorithm to minimize approximations (3.13), (3.14) (CCCP) requires calculating a gradient of the corresponding functions, its iterative nature being similar to **Algorithm 1**. In most cases, the nonconvex surrogate NSD performs as good as the $\log\text{-sum}$ approximations, and sometimes NSD happens to be better. Figure 4.7 shows that using functions (3.13), (3.14) yielded sparse controllers which are close or equal to optimal sparse controllers. In terms of optimality, among sparse controllers with the fixed number of zero rows the weighted 1∞-norm (3.12) is less efficient than nonconvex surrogates. This detail is important due to the fact that global convergence is not guaranteed.

On the ground of the results of the numerical experiments described above we arrived at the following conclusions:

- Using the 1∞-norm can lead to situations where the actual sparse-to-optimal performance drop α_{sp} is significantly smaller than the maximum admissible loss in performance $\hat{\alpha}$. Other surrogates usually do not suffer from this drawback.
- Using approximations to the matrix ℓ_0-quasinorm usually yields a sparse controller which is optimal among all controllers with the same number of zero rows.
- In some problems (e.g., see the example from section 4.2) the 1∞-norm does not succeed in the detection of sparse structure even for large values of $\hat{\alpha}$.
- The NSD function appears to be effective similarly to the nonconvex surrogate $\log\text{-sum}$, and in some cases the NSD happens to perform better.

5. CONCLUSION

In this paper we considered various approximations to the matrix ℓ_0-quasinorm, which can be applied to sparse feedback design in optimal control problems. The results of the numerical experiments show that to some extent all the surrogates analyzed can be applied to the detection of zero rows of the gain matrix, but nonconvex surrogates perform better.

Future efforts to be made in this area include the analysis of examples where one or another approximation to the matrix ℓ_0-quasinorm happens to be inefficient, and studying the alternative numerical procedures which can be applied to sparse feedback design, e.g., ADMM, alternating direction method of multipliers.

ACKNOWLEDGEMENTS

This work is supported by the Russian Science Foundation through grant No. 16-11-10015.

REFERENCES

1. Boyd, S., Ghaoui, L.E., Feron, E. & Balakrishnan, V. (1994) *Linear Matrix Inequalities in System and Control Theory.* Philadelphia, PA: SIAM Studies in Applied Mathematics.
2. Bykov, A., Shcherbakov, P. & Ding, M. (2016) A tractable nonconvex surrogate for the matrix ℓ_0-quasinorm: applications to sparse feedback design, *IFAC-PapersOnLine*, **49**(13), 53–58.
3. Candès, E.J., Wakin, M.B. & Boyd, S.P. (2008) Enhancing sparsity by reweighted ℓ_1 minimization, *Journal of Fourier Analysis and Applications*, **14**(5), 877–905.
4. Fazel, M., Hindi, H. & Boyd, S. (2003) Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices, *Proc. of American Control Conference*, 2156–2162.
5. Grant, M. & Boyd, S. (2014) *CVX: Matlab software for disciplined convex programming, version 2.1*, URL: http://cvxr.com/cvx.
6. Hartman, P. (1959) On functions representable as a difference of convex functions, *Pacific Journal of Mathematics*, **9**(3), 707–713.
7. Khlebnikov, M.V., Shcherbakov, P.S. & Chestnov V.N. (2015) Linear-quadratic regulator. I. a new solution, *Autom. Remote Control*, **76**(12), 2143–2155.
8. Khlebnikov, M.V. (2014) An ellipsoid approach to comparison of quadratic performance criteria, Stochastic optimization in computer science, **10**(1), 145–156. In Russian.
9. Lanckriet, G.R. & Rangarajan, B.K. (2009) On the convergence of the concave-convex procedure, *Advances in Neural Information Processing Systems*, **22**, 1759–1767.
10. Leibfritz, F. & Lipinski, W. (2003) *Description of the benchmark examples in COMPl$_e$ib 1.0*, URL: http://www.complib.de.
11. Levine, W.S. & Athans, M. (1970) On the determination of the optimal constant output feedback gains for linear multivariable systems, *IEEE Trans. on Automatic Control*, **15**(1), 44–48.
12. Lin, F., Fardad, M. & Jovanović, M.R. (2011) Augmented Lagrangian approach to design of structured optimal state feedback gains, *IEEE Trans. on Automatic Control*, **56**(12), 2923–2929.
13. Lin, F., Fardad, M. & Jovanović, M.R. (2013) Design of optimal sparse feedback gains via the alternating direction method of multipliers, *IEEE Trans. on Automatic Control*, **58**(9), 2426–2431.
14. Münz, U., Pfister, M. & Wolfrum P. (2014) Sensor and actuator placement for linear systems based on $\mathcal{H}_2$ and $\mathcal{H}_\infty$ optimization, *IEEE Trans. on Automatic Control*, **59**(11), 2984–2989.
15. Polyak, B., Khlebnikov, M. & Shcherbakov, P. (2013) An LMI approach to structured sparse feedback design in linear control systems, *Proc. of European Control Conference*, 833–838.
16. Polyak, B.T., Khlebnikov, M.V. & Shcherbakov, P.S. (2014) Sparse feedback in linear control systems, *Autom. Remote Control*, **75**(12), 2099–2111.
17. Polyak, B.T., Khlebnikov, M.V. & Shcherbakov, P.S. (2014) *Control of Linear Systems Subjected to Exogenous Disturbances: The Linear Matrix Inequality Technique.* Moscow, Russia: URSS. In Russian.
18. Sturm, J.F. (1999) Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optimization Methods and Software*, **11–12**, 625–653.
19. Toh, K.C., Todd, M.J. & Tutuncu, R.H. (1999) SDPT3 – a Matlab software package for semidefinite programming, *Optimization Methods and Software*, **11**, 545–581.
20. Vandenberghe, L. & Boyd, S. (1996) Semidefinite programming, *SIAM Review*, **38**(1), 49–95.
21. Yuille, A.L. & Rangarajan, A. (2003) The concave-convex procedure, *Neural Computation*, **15**, 915–936.
22. Zelazo, D., Schuler, S. & Allgöwer F. (2013) Performance and design of cycles in consensus networks, *Systems and Control Letters*, **62**(1), 85–96.

