University of Zielona Góra
Faculty of Electrical Engineering, Computer Science
and Telecommunications

Doctoral dissertation

Computational aspects in analysis and synthesis of repetitive processes

(Aspekty obliczeniowe analizy i syntezy procesów powtarzalnych)

Bartłomiej Sulikowski, M.Sc.

supervisor: Prof. Krzysztof Gałkowski, Ph.D., D.Sc.

This dissertation is partially supported by State /Poland/ Committee for Scientific Research, Grant no. 3 T11A 008 26.

Zielona Góra, 2005

Contents

List of Figures

Acknowledgments

I would like to express my deepest gratitude to my supervisor, professor Krzysztof Gałkowski from ISSI, UZ, for his support, help and patience during my research and writing this dissertation. Also, I'd like to thank professor Eric Rogers from the University of Southampton, UK, for the effective and fruitful cooperation and many fresh ideas and professor Dariusz Uciński from ISSI, UZ, for sharing with me his knowledge and experience in the optimization area. I owe some words to doctor Przemysław Jacewicz from ISSI, UZ for installing and managing the cluster, which was the crucial for the progress of an important part of my research.

Also, I'd like to thank my fiancee Karolina, for her love, patience and understanding; my parents, for their unconditional love – without them this adventure would not start at all; my brother Maciek and friends: Ewelina, Paweł, Magda, Wojtek, Maciek, Bartek and the last, but not the least, Witek, for their support during the hard times. Thank you all.

Shortcuts

1D	one-dimensional
2D	two-dimensional
nD	n-dimensional
	(multi-dimensional $n > 2$)
RM	2D Roesser model
FM	2D Fornasini-Marchesini model
LRP	Linear Repetitive Process
ILC	Iterative Learing Control
LMI	Linear Matrix Inequality
BMI	Bilinear Matrix Inequality
SDP	Semidefinite Programming
IPM	Interior Point Method
PI	Proportional plus Integral

Notation

R^n	n-dimensional real vector space
$\mathrm{R}^{n \times m}$	set of $n \times m$ real matrices
Z	set of integer numbers
$A > 0$ $(A < 0)$	symmetric positive (negative) defined matrix
$A \geq 0$ $(A \leq 0)$	symmetric semi-positive (semi-negative) defined matrix
$k > 0$ $(k < 0)$	positive (negative) scalar
$k \geq 0$ $(k \leq 0)$	non-negative (non-positive) scalar
A^T	transpose of the matrix
A^{-1}	inverse of the matrix
I_n	identity matrix of dimensions $n \times n$
$0_{n \times m}$	zero matrix of dimensions $n \times m$
$\mathrm{r}(A)$	spectral radius of the matrix
$\mathrm{diag}(A)$	diagonal of the matrix
$\mathrm{diag}(A_1, \ldots, A_n)$	a matrix with block matrices $A_1, \ldots, A_n$
	placed on diagonal
$\mathrm{trace}(A)$	trace of the matrix
$\det(A)$	determinant of the matrix
$(*)$	counterpart of the symmetric block in the matrix
$\lambda(A)$	an eigenvalues set of the matrix
$\lambda_{max}(A)$ $(\lambda_{min}(A))$	the maximal (minimal) eigenvalue of the matrix

Chapter 1

Introduction

For many years, there have been found many problems, which however solvable in theory, due to its complexity, have been unsolvable in practice. Nowadays, some of these problems can be solved due to the possibilities, which are provided by developing the modern algorithms and exploiting the huge computational power provided by the modern computers. The tasks of analysis and/or synthesis of the complex multidimensional dynamical systems have been included into that list of the practically unsolvable problems. Although there existed the strong theoretical results, there was not the appropriate methodology to solve those problems. The problems considered in this dissertation refer to the computer-aided analysis and synthesis of a special class of multidimensional systems — the so-called repetitive processes. In the dissertation, the methodology based on the results, which originally come from the area of the computer science, is developed and successfully applied to solve the considered problems. That include the appropriate application of the efficient numerical methods and developing algorithms to solve the stated tasks of analysis/synthesis. On the other hand, it is to note that schemes developed here for analysis/synthesis of the complex dynamical multidimensional systems, after the straightforward reformulation, can be used to provide the requested features of the algorithmic applications (e.g. the convergence of the iterative process).

Multidimensional (nD) systems are characterized by many (n) independent variables, on the contrary to classical systems, called here 1D, where there exists only one indeterminate (denoting most frequently time). Systems considered in this dissertation characterize of 2 independent variables. In the classical theory of 1D systems the independent variable used in the state–space description in most cases denotes the time (discrete or continuous), in 2D systems the independent variables can be treated as a vector time. In practice, for nD systems one of variables denotes the time and the rest of variables have the space meaning. They can be the space coordinates or the number of the current process phase, iteration or the trail. In general, in nD (2D) systems there exist the n (2) directions in which information propagate.

During last years nD systems have been found interesting from both theoretical and practical application standpoints. There can be found a number of books and papers regarding the considered class of systems (see e.g. [1, 2, 3, 4, 5, 6, 7] and references therein). Whenever the considered system is not suitable to be modeled using well known 1D models, 2D (nD) models are very strong alternative. nD (2D) systems have been found useful in modeling the physical processes in the areas of Control, Computer Science, Telecommunications, Acoustics, Electrical Engineering etc. The particular applications include nD filtering [8, 9], nD coding and decoding [10], image processing [11, 12] and multidimensional signal processing [13, 14, 15, 16].

Theory of nD systems is not a simple extension of the well known theory of 1D systems. There arise several obstacles and limitations related to the lack of the mathematical tools, or in the best situation, its very complicated forms. The reward for these obstacles is that the application of nD models in the description of the considered phenomena provides some new possibilities that were unavailable, when considering 1D models.

The special case of 2D systems are Linear Repetitive Processes (LRPs) [17, 18, 19, 20]. LRP is defined by a repetitive execution of an action, which lasts for a fixed finite duration. During each iteration (or, as it is called in context of that class of systems, a pass), an output, called the pass profile, is produced and it acts as a forcing function on the next pass profile. Hence there are two dynamics in the model of LRP, i.e. the first, which denotes the current pass number and the second, which denotes the position (time) during the pass.

The differences between LRPs and known 2D models are straightforward to see, since it is assumed that one of the independent variables used in the state-space model of LRPs is finite and the mixed discrete-continuous dynamics can be found (i.e. the variable denoting the number of pass is always discrete and the second regarding the position on the pass can be either discrete or continuous). In well known and widely used 2D state-space models (Roesser [21] or Fornasini Marchesini [22]) both variables are discrete and unbounded.

The applications of LRPs include long-wall coal cutting [17, 20], metal rolling [23, 17, 24, 25], Iterative Learning Control (ILC) schemes [26, 27, 28, 29] and iterative algorithms for solving nonlinear dynamic optimal control problems based on the maximum principle [30, 31]. Recently, the link between the spatially interconnected systems [32] and LRPs has been recognized and it seems that many of results obtained for LRPs can be adopted for that class of 2D (nD) systems.

Note here that ILC, as aforementioned, are one of the application area for LRP and can be treated as the tool for modeling the complex iterative numerical procedures [33, 34, 35]. In that sense Linear Repetitive Processes can be alone considered as strongly referring to the algorithmic theory. In essence, ILC can be treated as the schemes, where the control or, generally, the computation tasks are achieved iteratively [36, 28]. What is crucial here, the convergence of the iterative procedure is strongly related to the stability of the underlying LRP. Hence the synthesis of LRP can be considered as a tool to design the convergent iterative procedures and hence strongly links LRPs and algorithms area.

In general, one of the very crucial properties of the dynamic system is stability. For the considered class of dynamical systems, this dissertation deals with two basic types of stability, i.e. asymptotic stability and stability along the pass [17, 18]. Hence the emphasis here is put on the stability investigation (analysis) and the stabilization (synthesis) of LRPs. Due to the fact that LRPs are the distinct class of 2D systems, the analysis and synthesis tasks require to apply the appropriate methods, which, in general differ either from the methodology provided for the 1D classical systems ([37, 38] and references therein) and/or the results obtained for 2D systems [3, 7].

Another set of problems, which appear also in the classical 1D systems theory is the integration of the basic analysis/synthesis tasks with the stronger requirements. Those introduce the additional restrictions to the considered problem and hence can cause the additional problems from both, theoretical (formulating the appropriate conditions) and practical (numerical problems), standpoints. The mentioned aspects considered in this dissertation include topics related to stability margins and developing the approach for the so-called model matching. The special subclass of the synthesis tasks is addressed to the output based stabilization schemes. Note that those supplementary topics are always considered as ones introducing the additional constraints to the basic problems. Frequently, it is necessary "to pay" for solving such extended problems with decreasing the area of possible solutions and, in many cases, increasing the numerical effort to obtain the solution. What is more, the additional aspects, which are considered beyond the basic stability of LRP (e.g. the stability margins or governing the requested form of the closed loop system), re-mapped to the ILC ensure, beyond the convergence, faster obtaining the requested result, e.g. the faster minimizing the tracking error. Hence the methods developed originally to check/ensure the stability of the dynamical system can be applied for improving the behavior of the iterative process.

As aforementioned, it is necessary to realize that considered problems of analysis/synthesis of LRPs can cause serious problems from the theoretical and numerical standpoints. Despite the fact that several conditions regarding the stability of LRPs have been presented e.g. in [17, 39, 40, 18], there exist the serious limitations in application of those conditions, since they deal mainly with the 2D transfer function. Since for 2D system the poles of a transfer function are the curves on the complex plain (not isolated points as for the 1D case), there arise serious difficulties with the stability analysis and stabilization for this class of systems. Due to this, the new, efficient methodology for analysis and synthesis of LRPs is required. Hence it is the natural question to consider, if the problems of analysis and/or synthesis, which appear in the area of LRPs (and in general 2D systems), can be formulated in the way to assure the final effective solution. Unfortunately, the majority of those problems, considered in a classic way, are considered as being $\mathcal{NP}$-hard [41]. It is due to the fact that original sufficient and necessary conditions for analysis require in fact performing the infinite number of of classic (1D) stability tests, which appears to be impossible. The situation is even worse in the synthesis task for multivariable (MIMO) LRPs. One of the possible solutions for this crucial problem comes from the Lyapunov theory, strengthened by the fact that there appeared recently the efficient numerically methods of the Linear Matrix Inequalities (LMIs) [42, 43, 44, 45], which are based on the Interior Point Methods convex optimization algorithms.

For 2D (nD in general) systems, LMIs turn out to provide in fact almost only efficient numerically method (solution provided in the polynomial time) and also an easy and natural extension to stabilization has appeared (see e.g. [46, 47, 48, 49, 50]). What is important to underline that for many cases, when the analysis/synthesis of LRPs (and nD systems in general) is considered, the application of LMIs is one of the very limited number of choices to handle efficiently those problems. Another reported in the literature possibility is the so-called μ-analysis approach ([51]), which however is more complicated, but still effective for solving these complex problems. Here, the attention is limited on using the LMI methods.

Thanks to the application of the LMI methods and, what follows, exploiting its numerical efficiency, the subclass of $\mathcal{NP}$-hard problems is possible to be approximated with the application of the Lyapunov based methodology and then solved by the polynomial time algorithms. The relevance of such a reformulation cannot be underestimated. It is to note that due to its numerical efficiency, LMIs assure that the original $\mathcal{NP}$-hard problem, reformulated into the approximated valid $\mathcal{P}$-class problem, can be solved using very efficient algorithms. The drawback of the application the LMIs in the analysis and synthesis of LRPs is recognized as that the conditions defined in terms of LMIs for 2D systems (LRPs) are only sufficient. Nevertheless, the method how to lower the conservativeness of those conditions and finally get closer to the necessary and sufficient LMI conditions for stability of 2D systems has been presented recently in [52]. The application of these conditions involve the rapid increase of the size and the complexity of the problem. The similar approach after the appropriate modifications can be applied for the analysis of LRPs.

Note that even if the analysis and/or the synthesis of LRPs can be shown to be polynomial time solvable, those tasks can cause serious problems regarding the numerics. It is especially apparent, when the considered systems are highly dimensioned. Then even the application of the polynomial time method can be insufficient to provide the solution accurately. This aspect becomes visible when taking into account the fact that those problems are solved using computers. In view of computer-aided analysis/synthesis the following topics appear: storing the data describing the considered problem in the memory, performing the computations according to considered analysis/synthesis tasks and finally simulating the system. The second point is the most demanding one. Hence to provide the adequate computational power for the efficient solving the analysis/synthesis problems the computer clusters have been used. It is to note that due to the fact that the considered topics are highly specialist, there are not available the appropriate software packages allowing to solve those kinds of problems directly. Hence the method of how to reformulate the analysis/synthesis problems into the form solvable by existing cluster software can be treated as the original and practical result of this dissertation.

In the dissertation various subclasses of LRPs are considered. Except of the basic LRPs (discrete and differential), first defined in [17], the generalized model of LRP is considered ([53, 54]). As a special subclass of LRPs the extended model has been considered in [55] and the wave LRP – defined in [56]. There also have been defined the singular LRPs ([57, 58]). In general, for that whole set of models, it is hard to define the applicable conditions towards stability along the pass. Hence in many cases, it is possible to settle only for asymptotic stability. Dealing with this property is based on the investigation of the so-called 1D equivalent model of LRP ([59, 60, 61, 40, 25]). The application of the analysis/synthesis methods for the 1D equivalent model can cause serious difficulties, since it is based on the possibly highly dimensioned model. In the sequel, some remedies for these problems have been presented. One possible solution here is the simplification of the structure of the considered model. Another one is using the computer clusters of large computational power to solve the stated problems.

An important question arises, when considering the application of the clusters in service to solve problems of analysis/synthesis of LRPs. Since the problems considered in this dissertation are high dimensioned and finally are presented as the convex optimization problems over huge numbers of variables, two aspects appear. These are: a memory complexity and a computational complexity. The first one can be easily determined basing on the size of the problem (size of the LMI) and the number of variables. It is natural that the memory complexity increases significantly when the size of the problem increases, however here the obstacles connected with this fact have been overcome by simply adding the RAM memory or increasing the size swap disk on the computer/cluster on which the simulations are made. Hence in the sequel, the memory complexity aspects are not considered.

The special emphasis is put on the computational complexity aspects. Since the problems of analysis/synthesis are presented as LMIs, it is known that they can be solved using the polynomial time algorithms. Those results regard both: asymptotic stability and stability along the pass. In the case of asymptotic stability, the cluster provides the computational power necessary to solve the huge problems, unsolvable on the single PC. However, those problems still have to be considered as the polynomial time problems. For stability along the pass, the comparison between two existing LMI conditions is provided regarding the applicability of those conditions, the growth of time required to solve the problem in function of its size and the study regarding the level of conservativeness of both conditions.

In the practical applications, the question about controlling the LRPs with the appropriate (required) performance appears. There have been published some preliminary results regarding the stability aspects, but here the synthesis problem governing the additional properties ("beyond" the stability) of the system in the closed loop configuration, e.g. ensuring the prescribed stability margins or the assurance of the prescribed form of the closed loop system, are considered. The further extensions include the application of the developed schemes in the practical applications, where the natural goals to be achieved can be defined as driving the considered system to the required output (called the reference signal) and the disturbance rejection.

Concluding, two-fold relationships of the reported results to the computer science area can be pointed out, i.e. 1st – LRPs alone have links to the algorithms area as described above; and 2nd – effective use of this approach involves strong numerical problems, solving of which is the subject of this work.

Thesis

Due to the aforementioned facts, the following leading thesis of this dissertation is proposed:

> ***It is possible to develop numerically effective analysis and synthesis methods for complex high dimensional repetitive processes, based on the use of LMI schemes, strengthened by the modern numerical tools involving parallel computing.***

and in the sequel the results proving that thesis are presented.

Regarding the contribution of the results presented in this dissertation to the "state of art", it should be underlined that it can be seen from both, theoretical and practical, points of view. The theoretical aspects covered in this dissertation include:

- definitions and descriptions of the considered systems,
- the review of existing conditions analysis and synthesis,
- determining the LMI conditions towards stability and controller design, additionally regarding the requested dynamical properties of the system,
- developing the control schemes towards the required performance.

The practical aspects include:

- implementations of the functions enabling to check the considered conditions,
- developing the methodology allowing to apply the parallel computers (clusters) to solve the analysis/synthesis problems,
- tests and simulations of the considered control schemes.

The framework of this dissertation can be presented as follows:

Chapter 2 contains the background information on 2D systems and LRPs. The basic models of considered LRPs, the boundary condition types and the relations between them and classical 2D systems are provided here. Also the existing analysis/synthesis conditions for LRPs are presented. This chapter finishes with the list of selected real-live systems which can be (are) modeled with LRP models.

Chapter 3 contains the basic information regarding the LMIs and SDPs and their applications in the control theory. Also the Interior Point Method convex optimization algorithms, which are used to solve problems (including the parallelized version) are described. *Chapter 3* provides the list of available LMI/SDP software packages and the description of the selected packages (those, which are used in the computational examples). At the end of this chapter, very short note on the clusters (parallel computing) and the chosen SDP solver is included.

In *Chapter 4* the LMI approach for analysis and synthesis of the considered classes of LRPs is provided. Some simulation results (those regarding solving huge numerical problems) presented in *Chapter 4* have been obtained with application of the parallel computing. In this chapter two types of stability, i.e. asymptotic and along the pass are considered. The integration of the stability/stabilization conditions with the additional requirements regarding the dynamical properties of the closed loop system are also presented here. Those requirements include: computing the stability margins and application of the model matching. In this chapter the output-based control schemes are introduced as well. *Chapter 4* presents also the methods how to deal with large (huge) dimensioned problems which appear in the analysis/synthesis tasks. The list of those approaches include: the application of the parallel computing, simplification of the stated problem by decoupling of the dynamics and finally, exploiting the features of the iterative approach – successive stabilization algorithm.

Chapter 5 addresses the task of the control for performance of LRPs. Here, the main goal is to provide the method, which allows to achieve the required reference signal after the sufficient (small) number of passes and the disturbance rejection, with the particular attention paid to the application of the developed control schemes.

The dissertation finishes with Appendices, presenting the description how to use the selected LMI/SDP solvers (Appendix A) and containing selected functions used in simulation examples presented in the dissertation (Appendix B).

To highlight the computational aspects of presented in the dissertation topics, examples are provided. They are assumed to present all necessary data, which appear during the computations to explain how the presented conditions are performed.

Topics which are not covered in this dissertation, but are planed as the further problems to be solved, include the models with uncertainties and related to this the so-called robust control. It is to note that these problems are the subject of the on-going work in which author of this dissertation is also involved (see e.g. [7, 46, 48, 50, 20]).

In the remainder of this introduction, it to be written that most of the results presented in this dissertation have been achieved, when working in the large international research group consisting Zielona Góra (prof. Gałkowski), Southampton - UK (prof. Rogers) and Sheffield -

UK (Prof. Owens) – see the references. Hence the most of the Author results is published in the papers with four coauthors, but in the best journals and in the most known international conference proceedings. Finally, it is to note that these results are mostly achieved by the author of this thesis and hence can be included here.

Chapter 2

Two-dimensional (2D) state-space models and repetitive processes

Last three decades provide very rapid development of 2D systems theory and applications based upon them. In the area of the automatic control the most common used state-space models are the 2D Roesser (RM) model (see [21]) and the 2D Fornasini-Marchesini (FM) model (see [22]). Note that similarly to the classical 1D state-space models, there can be distinguished between the state and the output equations as well. An another, distinct sub-class of two-dimensional systems are Linear Repetitive Processes (LRP) (for references see e.g. [17, 62, 40, 50]). The research results presented in this dissertation are maintained just for LRPs. However, to put things in order, first there are presented the 2D Roesser model and the Fornasini-Marchesini model.

2.1 2D state-space models

2.1.1 Roesser model (RM)

The 2D Roesser model [21] is defined as follows

$$\begin{aligned} \begin{bmatrix} x^h(i+1,j) \\ x^v(i,j+1) \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x^h(i,j) \\ x^v(i,j) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(i,j), \\ y(i,j) &= \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x^h(i,j) \\ x^v(i,j) \end{bmatrix} + Du(i,j), \end{aligned} \tag{2.1}$$

where

$i, j \in \mathrm{Z}_+$ - vertical and horizontal indeterminates, represent direction, respectively,

$x^h(i,j) \in \mathrm{R}^{n_1}$ - the local horizontal state subvector,

$x^v(i,j) \in \mathrm{R}^{n_2}$ - the local vertical state subvector,

$x(i,j) = \begin{bmatrix} x^h(i,j) \\ x^v(i,j) \end{bmatrix} \in \mathrm{R}^{n_1+n_2}$ - the local state vector,

$u(i,j) \in \mathrm{R}^r$ - the input vector,

$y(i,j) \in \mathrm{R}^m$ - the output vector,

$A_{11},\ A_{12},\ A_{21},\ A_{22},\ B_1,\ B_2,\ C_1,\ C_2,\ D$ - matrices of appropriate dimensions.

The boundary conditions for (2.1) in their simplest (static) form are defined as:

$$\begin{cases} x^h(0,j) = x^h_{0j} \in \mathrm{R}^{n_1}, & j \in \mathrm{Z}_+, \\ x^v(i,0) = x^v_{i0} \in \mathrm{R}^{n_2}, & i \in \mathrm{Z}_+. \end{cases} \tag{2.2}$$

In the above model the explicit distinction between the vertical and horizontal parts of the state vectors are given. Note that using the above state-space description there arise very strong connections with the polynomial matrix theory. Hence with the application of the appropriate delay operators, defined as follows

$$x(i,j) := z_1 x(i,j+1),\ \ x(i,j) := z_2 x(i+1,j),$$

it is possible to obtain the following transfer function for the considered RM of (2.1)

$$G_{\mathrm{RM}}(z_1,z_2) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \left(\begin{bmatrix} I - z_1 A_{11} & -z_1 A_{12} \\ -z_2 A_{21} & I - z_2 A_{22} \end{bmatrix} \right)^{-1} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} + D. \tag{2.3}$$

Hence the characteristic polynomial of RM is given by

$$\mathcal{C}_{\mathrm{RM}} = \det \left(\begin{bmatrix} I - z_1 A_{11} & -z_1 A_{12} \\ -z_2 A_{21} & I - z_2 A_{22} \end{bmatrix} \right). \tag{2.4}$$

2.1.2 Fornasini-Marchesini model (FM)

The 2D Fornasini-Marchesini model [22] is defined as follows

$$\begin{aligned} x(i+1,j+1) &= A_1 x(i+1,j) + A_2 x(i,j+1) + B_1 u(i+1,j) + B_2 u(i,j+1), \\ y(i,j) &= C x(i,j) + D u(i,j), \end{aligned} \tag{2.5}$$

where

$i,j \in \mathrm{Z}_+$ - vertical and horizontal indeterminates, represent direction, respectively,

$x(i,j) \in \mathrm{R}^n$ - the local state vector,

$u(i,j) \in \mathrm{R}^r$ - the input vector,

$y(i,j) \in \mathrm{R}^m$ - the output vector,

$A_1,\ A_2,\ B,\ C,\ D$ - matrices of appropriate dimensions.

The boundary conditions for (2.5) in their simplest (static) form are defined as:

$$\begin{cases} x(0,j) = x_{0j} \in \mathrm{R}^n, & j \in \mathrm{Z}_+, \\ x(i,0) = x_{i0} \in \mathrm{R}^n, & i \in \mathrm{Z}_+. \end{cases} \tag{2.6}$$

Note that here the vertical and horizontal components are not given explicitly as it had a place in the RM case.

For FM defined by (2.5), the transfer function is defined as

$$G_{\mathrm{FM}}(z_1, z_2) = C\left(I - z_1 A_2 - z_2 A_1\right)^{-1}\left(z_1 B_2 + z_2 B_1\right) + D \tag{2.7}$$

and the characteristic polynomial is defined as

$$\mathcal{C}_{\mathrm{FM}} = \det\left(I - z_1 A_2 - z_2 A_1\right). \tag{2.8}$$

Remark 2.1 *In general, due to the particular requirements many extensions of the above models was defined. The number of them can be found in e.g. [3, 63, 64] or [65].*

Remark 2.2 *Note that RM and FM provide the natural and easy way to extend the model from 2D to nD in general.*

2.2 Linear Repetitive Processes

Linear Repetitive Processes (LRPs) are a very important class of 2D systems, especially from the practical point of view. The models of LRPs describe the repetitive execution of some action which lasts for the fixed finite duration. Such a single execution can be treated as an iteration of some complex process, but in the context of LRP, it is called the pass. Hence there are two distinct dynamics in the model of LRP, i.e. along the pass and from pass to pass. The features that differ LRPs from the usual 2D systems (in RM or FM form) are that the indeterminate regarding the dynamics in the along the pass direction is limited and that the position on the chosen pass can be either discrete and then denoted by $p \in \mathrm{Z}_+$, or continuous, denoted by $t \in \mathrm{R}_+ \cup \{0\}$. In those cases the discrete, or respectively, differential LRPs are considered. The pass length is denoted by α and the current pass number by k. Note that in both cases, the pass number k is always a discrete number. This classification has been changed lately slightly when so-called multidimensional hybrid systems (see e.g. [66]) have been introduced.

Due to the fact that LRPs indeed are the 2D systems, the control schemes known for the classical 1D systems, when applied here, can fail. It is because, those schemes do not regard the implicit 2D structure of the considered system. Even, if it is possible to employ the 1D control approach for the every single pass (treated as separate 1D systems), it does not concern the influence that comes form the previous pass. Hence the output vector of LRP (called the pass profile vector) values can increase to the enormous values of the amplitude and/or contain the oscillations. Due to that, it is necessary to develop the dedicated approach to control the considered class of systems.

To define formally a LRP, for the assumed constant pass length denoted by $\alpha < +\infty$, the pass profile $y_k(p)$, $0 \le p \le \alpha - 1$, for the discrete case ($y_k(t)$, $0 \le p \le \alpha - 1$, for the differential case), generated on pass k acts as a forcing function the next pass profile $y_{k+1}(p)$, $0 \le p < \alpha - 1$, $k \ge 0$ ($y_{k+1}(t)$, $0 \le p < \alpha - 1$, $k \ge 0$).

The models of LRPs are closest in their form to 2D Roesser models. The first equation in the LRP models is called the state equation and the second - the pass profile (an output) equation. The principal difference between 2D RM and LRP is that in the LRP model the state vector $x_k(p)$ can be treated as an equivalence of the horizontal subvector $x^h(i, j)$ of the RM and of the pass profile vector $y_k(p)$ represents the output of the model and the vertical subvector $x^v(i, j)$ of the RM simultaneously.

2.2.1 Discrete LRPs

Following [17, 18], the state-space model of a discrete linear repetitive process has the following form over $0 \leq p \leq \alpha - 1,\ k \geq 0$

$$x_{k+1}(p+1) = Ax_{k+1}(p) + B_0 y_k(p) + Bu_{k+1}(p), \tag{2.9}$$
$$y_{k+1}(p) = Cx_{k+1}(p) + D_0 y_k(p) + Du_{k+1}(p), \tag{2.10}$$

where

$0 \leq p \leq \alpha - 1\ \in \mathrm{Z}_+$ - the discrete position on the current pass,

$k \in \mathrm{Z}_+$ – the current pass number,

$x_k(p) \in \mathrm{R}^n$ – the state vector,

$y_k(p) \in \mathrm{R}^m$ – the pass profile (output) vector,

$u_k(p) \in \mathrm{R}^r$ – the input vector,

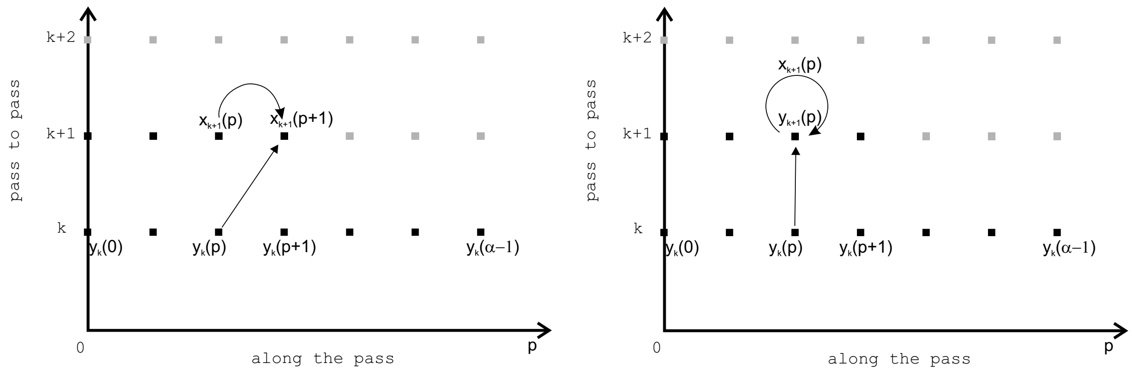
$A,\ B_0,\ C,\ D_0,\ B,\ D$ – matrices of appropriate dimensions.

To complete the process description, it is necessary to specify the 'initial conditions' - termed the boundary conditions here, i.e. the state initial vector on each pass and the initial pass profile. The simplest possible form for these is

$$\begin{aligned} x_{k+1}(0) &= d_{k+1},\ k \geq 0, \\ y_0(p) &= f(p), \end{aligned} \tag{2.11}$$

where $d_{k+1} \in \mathrm{R}^n$ - vector and the entries in the vector $f(p) \in \mathrm{R}^m$ are known functions of p.

Figure 2.1 illustrates the updating structure of the state and pass profile vectors in (2.9)-(2.10).



a) b)

Figure 2.1. Illustrating the state (a) and pass (b) profile vector updating structure in (2.9)-(2.10)

The characteristic polynomial for (2.9)-(2.10) is defined as follows

$$\mathcal{C}_{\text{discreteLRP}} = \det\left(\begin{bmatrix} I - z_1 A & -z_1 B_0 \\ -z_2 C & I - z_2 D_0 \end{bmatrix}\right), \tag{2.12}$$

where $z_1, z_2 \in \mathrm{C}$ are the inverses of the z-transform variables in the horizontal and vertical directions respectively. They can be also considered as the unit delay operators in those directions. They are defined as follows

$$x_k(p) := z_1 x_k(p+1), \;\; x_k(p) := z_2 x_{k+1}(p). \tag{2.13}$$

The 2D nature of LRPs allows considering the hybrid processes, i.e. discrete-continuous, which is presented in the next section.

2.2.2 Differential LRPs

The differential LRP has the following form over $0 \le t < \alpha,\; k \ge 0$

$$\dot{x}_{k+1}(t) = Ax_{k+1}(t) + B_0 y_k(t) + Bu_{k+1}(t), \tag{2.14}$$
$$y_{k+1}(t) = Cx_{k+1}(t) + D_0 y_k(t) + Du_{k+1}(t), \tag{2.15}$$

where

$0 \le t < \alpha \;\in \mathrm{R}_+ \cup \{0\}$ – the continuous position on the current pass,

$k \in \mathrm{Z}_+$ – the current pass number,

$x_k(t) \in \mathrm{R}^n$ – the state vector,

$y_k(t) \in \mathrm{R}^m$ – the pass profile (output) vector,

$u_k(t) \in \mathrm{R}^r$ – the input vector,

$A,\; B_0,\; C,\; D_0,\; B,\; D$ – matrices of appropriate dimensions.

It is clear that the process is continuous along the pass and discrete from pass to pass.

Again, to complete the process description, it is necessary to specify the 'initial conditions' - termed the boundary conditions here, i.e. the state initial vector on each pass and the initial pass profile. The simplest possible form for these is

$$\begin{aligned} x_{k+1}(0) &= d_{k+1},\; k \ge 0, \\ y_0(t) &= f(t), \end{aligned} \tag{2.16}$$

where $d_{k+1} \in \mathrm{R}^n$ are known vectors and the $f(t) \in \mathrm{R}^m$ is the vector valued function, which generates an appropriate $f(t) \in \mathrm{R}^m$ for given t .

The characteristic polynomial for (2.14)-(2.15) is defined as follows

$$\mathcal{C}_{\mathrm{diffLRP}} = \det\left(\begin{bmatrix} sI - A & -B_0 \\ -zC & I - zD_0 \end{bmatrix}\right), \tag{2.17}$$

where $s \in \mathrm{C}$ is the Laplace transform indeterminate and $z \in \mathrm{C}$ comes as before form the use of the z-transform in the direction form pass to pass.

There was some research work (see e.g. [67, 68]), where the definition and the influence of the dynamical boundary conditions for LRPs (discrete and differential) have been considered in details, however these results are not given here, due to the fact that the sequel of this dissertation does not regard those topics.

Remark 2.3 *The same notation for those two types of processes, i.e. discrete and differential is used but in every place, when the confusion may occur, additional statement is provided.*

For the purpose of the sequel requirements, define so-called: 2D system plant and 2D extended input matrix of considered models of LRPs as follows

$$\Upsilon = \begin{bmatrix} A & B_0 \\ C & D_0 \end{bmatrix}, \; \Omega = \begin{bmatrix} B \\ D \end{bmatrix}. \tag{2.18}$$

2.2.3 Generalized discrete LRPs

Note that the basic models of the discrete LRP presented in the subsection 2.2 assume that the state and the pass profile at the given position depend only on the state "delayed" along the pass and the previous pass profile. However, this single step influence can be extended by the quite natural completion a set of past profile vectors. Such a extension of the model can be motivated by the practical applications e.g. the robotic systems ([69]) or in the Iterative Learning Control ([28, 29]), where in some particular cases more information from the past influence the dynamics in the system. The model of the discrete LRP with one additional factor has been considered in [55]. The full set of previous pass profile vector influence has been considered in [54, 70]. The state-space model of a so-called generalized discrete LRP is defined by the following equations over $0 \le p \le \alpha - 1,\ k \ge 0$,

$$x_{k+1}(p+1) = Ax_{k+1}(p) + Bu_{k+1}(p) + \sum_{j=0}^{\alpha-1} B_j y_k(j), \tag{2.19}$$

$$y_{k+1}(p) = Cx_{k+1}(p) + Du_{k+1}(p) + \sum_{j=0}^{\alpha-1} D_j y_k(j), \tag{2.20}$$

where all vectors and matrices have the same meaning as for (2.9)-(2.10). The boundary conditions for this model are of the form (2.11).

Motivation for considering processes of the form (2.19)-(2.20) arises from applications, where the current pass profile at any point along the pass is a function of more than one point on the previous pass. Clearly, the process of (2.19)-(2.20) is not upper right quadrant casual (in the 2D systems sense) since the point $(k+1, p)$ is influenced also by the collection of points $(k, p+1), (k, p+2), \ldots, (k, \alpha-1)$. For the causality requirement, it would be required that $B_j = 0,\ D_j = 0,\ \ \forall j > p$.

Suppose now that $\forall p = 0, 1, \ldots, \alpha - 1$

$$B_j = \begin{cases} B_0, & j = p \\ 0, & j \neq 0 \end{cases} \quad \text{and also} \quad D_j = \begin{cases} D_0, & j = p \\ 0, & j \neq 0 \end{cases}.$$

Then the model of (2.19)-(2.20) reduces to (2.9)-(2.10), where it is assumed that the current state and pass profile vectors are only directly influenced by the pass profile vector at the same point on the previous pass.

Figure 2.2 illustrates the updating structure of the state and pass profile vectors in (2.19)-(2.20) (which clearly includes that of (2.9)-(2.10) as a special case).

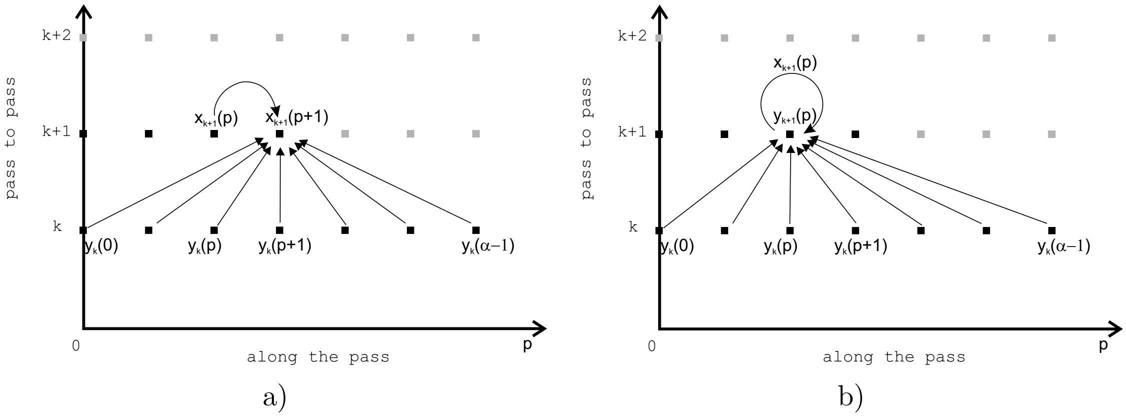


Figure 2.2. Illustrating the state (a) and pass profile (b) vector updating structure in (2.19)-(2.20)

2.3 Equivalent 1D model for discrete LRP

For the discrete LRPs described by the state-space models (2.9)-(2.10) or (2.19)-(2.20), it has turned out that a powerful approach to some control related problems is to exploit their inherent 2D linear systems structure and, in effect, adapt tools/results first developed for 2D linear systems described by the extensively studied Roesser and Fornasini Marchesini state-space models. In cases, where this approach is not applicable, e.g. pass controllability [71] or the presence of so-called dynamic boundary conditions [67, 68] which have no Roesser or Fornasini Marchesini model equivalents, 1D equivalent model has provided the analysis basis on which to solve the problems being considered.

Here, it is clear that the 2D systems approach does also not prove to be a suitable setting for analysis of key systems theoretic properties. The construction of the 1D equivalent model for discrete LRPs consists of the following steps (they are here presented for the basic model (2.9)-(2.10) — [72, 62], since the construction for the generalized LRP is done actually in the same manner — [25]).

The first step is introducing the following substitutions into (2.9)-(2.10)

$$l = k + 1, \;\; v_l = y_{l-1}. \tag{2.21}$$

Then the considered model of discrete LRP becomes

$$\begin{aligned} x_l(p+1) &= Ax_l(p) + B_0 v_l(p) + Bu_l(p), && (2.22)\\ v_{l+1}(p) &= Cx_l(p) + D_0 v_l(p) + Du_l(p). && (2.23) \end{aligned}$$

Next, introduce the so-called global state, input and pass profile vectors (termed supervectors) as

$$X(l) = \begin{bmatrix} x_l(1) \\ x_l(2) \\ x_l(3) \\ \vdots \\ x_l(\alpha) \end{bmatrix}, \; U(l) = \begin{bmatrix} u_l(0) \\ u_l(1) \\ u_l(2) \\ \vdots \\ u_l(\alpha-1) \end{bmatrix}, \; V(l) = \begin{bmatrix} v_l(0) \\ v_l(1) \\ v_l(2) \\ \vdots \\ v_l(\alpha-1) \end{bmatrix},$$

where $X(l) \in \mathrm{R}^{n\alpha}$, $Y(l) \in \mathrm{R}^{m\alpha}$, $U(l) \in \mathrm{R}^{r\alpha}$. Then the 1D equivalent model for the dynamics of discrete LRPs described by (2.9)-(2.10) is

$$\begin{aligned} X(l) &= \Gamma V(l) + \Sigma U(l) + \Psi_0 d_l, \quad (2.24)\\ V(l+1) &= \Phi V(l) + \Xi U(l) + \Theta_0 d_l, \quad (2.25)\end{aligned}$$

where

$$\Gamma = \begin{bmatrix} B_0 & 0 & 0 & \dots & 0 \\ AB_0 & B_0 & 0 & \dots & 0 \\ A^2B_0 & AB_0 & B_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{\alpha-1}B_0 & A^{\alpha-2}B_0 & A^{\alpha-3}B_0 & \dots & B_0 \end{bmatrix}, \quad (2.26)$$

$$\Sigma = \begin{bmatrix} B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ A^2B & AB & B & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{\alpha-1}B & A^{\alpha-2}B & A^{\alpha-3}B & \dots & B \end{bmatrix}, \ \Psi_0 = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^{\alpha} \end{bmatrix},$$

$$\Xi = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{\alpha-2}B & CA^{\alpha-3}B & CA^{\alpha-4}B & \dots & D \end{bmatrix}, \ \Theta_0 = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\alpha-1} \end{bmatrix},$$

$$\Phi = \begin{bmatrix} D_0 & 0 & 0 & \dots & 0 \\ CB_0 & D_0 & 0 & \dots & 0 \\ CAB_0 & CB_0 & D_0 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ CA^{\alpha-2}B_0 & CA^{\alpha-3}B_0 & CA^{\alpha-4}B_0 & \dots & D_0 \end{bmatrix}.$$

Applying the above procedure, it is straightforward to see that the 1D equivalent model for the generalized LRP of (2.19)-(2.20) becomes also in the form of (2.24)-(2.25), where all equivalent 1D model matrices are of the structure (2.26) except Φ and Γ, which take the separate forms of (for the detailed description see [54, 70])

$$\Phi = \begin{bmatrix} D_0 & D_1 & D_2 & \dots & D_{\alpha-1} \\ CB_0+D_0 & CB_1+D_1 & CB_2+D_2 & \dots & CB_{\alpha-1}+D_{\alpha-1} \\ \sum_{i=0}^{1} CA^iB_0+D_0 & \sum_{i=0}^{1} CA^iB_1+D_1 & \sum_{i=0}^{1} CA^iB_2+D_2 & \dots & \sum_{i=0}^{1} CA^iB_{\alpha-1}+D_{\alpha-1} \\ \sum_{i=0}^{2} CA^iB_0+D_0 & \sum_{i=0}^{2} CA^iB_1+D_1 & \sum_{i=0}^{2} CA^iB_2+D_2 & \dots & \sum_{i=0}^{2} CA^iB_{\alpha-1}+D_{\alpha-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{\alpha-2} CA^iB_0+D_0 & \sum_{i=0}^{\alpha-2} CA^iB_1+D_1 & \sum_{i=0}^{\alpha-2} CA^iB_2+D_2 & \dots & \sum_{i=0}^{\alpha-2} CA^iB_{\alpha-1}+D_{\alpha-1} \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} B_0 & B_1 & B_2 & \dots & B_{\alpha-1} \\ B_0 + AB_0 & B_1 + AB_1 & B_2 + AB_2 & \dots & B_{\alpha-1} + AB_{\alpha-1} \\ \sum_{i=0}^{2} A^i B_0 & \sum_{i=0}^{2} A^i B_1 & \sum_{i=0}^{2} A^i B_2 & \dots & \sum_{i=0}^{2} A^i B_{\alpha-1} \\ \sum_{i=0}^{3} A^i B_0 & \sum_{i=0}^{3} A^i B_1 & \sum_{i=0}^{3} A^i B_2 & \dots & \sum_{i=0}^{3} A^i B_{\alpha-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{\alpha-1} A^i B_0 & \sum_{i=0}^{\alpha-1} A^i B_1 & \sum_{i=0}^{\alpha-1} A^i B_2 & \dots & \sum_{i=0}^{\alpha-1} A^i B_{\alpha-1} \end{bmatrix}. \tag{2.27}$$

In the 1D equivalent model, only equation (2.25) is dynamical along the pass number l. It is easy to see that (2.25) plays the role of the state equation in this model and only this equation imposes the stability of the considered system. The equation (2.24) is now static, and indeed it can be treated as a state observer equation. Nevertheless, there still exists dynamics along the pass (along p), however it is hidden in the structure of the 1D equivalent model matrices and definitions of new vector variables $X(l)$, $V(l)$ and $U(l)$.

Remark 2.4 *It is straightforward to see that the pass length in the discrete LRP depends on the discretization period of the differential LRP. In what follows, the better precision required, the larger pass length α obtained. The detailed description of those topics can be found e.g. in [62, 73, 74, 75]. Due to that fact of the possible large α, the obtained 1D equivalent model of LRP can be of the huge dimensions. This comes from the fact that in 1D equivalent model, each vector or matrix is of the dimensions of the pass length multiplied by the appropriate size.*

Example 2.1 *To highlight the possible huge dimensionality and problems that appears, when the 1D equivalent model of LRP is used, consider for instance, the 2D LRP where:*

the pass length, $\alpha = 200$,

the number of the states, $n = 12$,

the number of the outputs, $m = 5$,

the number of inputs, $r = 7$.

Then the 1D model vectors and matrices are of the following dimensions

$X(l) \in \mathrm{R}^{2400}$, $V(l) \in \mathrm{R}^{1000}$, $V(l+1) \in \mathrm{R}^{1000}$ *and* $U(l) \in \mathrm{R}^{1400}$,

$\Phi \in \mathrm{R}^{1000\times1000}$, $\Delta \in \mathrm{R}^{1000\times1400}$, $\Theta_0 \in \mathrm{R}^{1000\times10}$, $\Gamma \in \mathrm{R}^{2400\times2400}$, $\Sigma \in \mathrm{R}^{2400\times1400}$ *and finally* $\Psi_0 \in \mathrm{R}^{2400\times10}$.

These data stored in the RAM memory of the personal computer (assuming that the every single entry has to be allocated on eight bytes since it is treated as a double real number in the way that e.g. Matlab *does it) take in total* 65652800 *bytes (taking* 1 *kB as* 1024 *bytes and* 1 *MB as* 1024 *kB, it gives* $\approx$ 64114 *kB or* $\approx$ 62.5 *MB) of the "spare" memory required for storing the above variables only. There are additional requirements, when taking into account the fact that those matrices are to be processed. Note that at the moment personal computers are equipped with* 256 *MB of RAM (or less) on usual.*

It is important to mention here that the construction of the 1D equivalent model is possible only for discrete LRPs. For differential LRPs of (2.14)-(2.15) it is impossible to perform due to the fact that there are no discrete steps in the along the pass direction. Hence the construction of supervectors and 1D equivalent model matrices is impossible.

2.4 Analysis and synthesis of LRPs

When considering the dynamical system, the very first thing taken into account is the stability of that system. The same notion appears, when investigating the properties of LRPs. Then the basic problems are: how to define the stability, how to test it (analysis), and finally, what to do, when the system has been affirmed to be unstable (synthesis). On the contrary to 1D classical systems (either discrete of continuous time), for 2D (nD) systems the analysis/synthesis problems are sophisticated and to solve require application of efficient methods. The analysis itself is required to determine, if the tested 2D system (LRP) can be left without any external input (so-called free evolution of the system) and the synthesis task has two main goals, i.e. to drive the unstable system to stability and/or to ensure the required performance of the controlled system.

A stability theory [17] for LRPs is based on an abstract model of the process dynamics in a Banach space (here denoted by E_α) of the form

$$y_{k+1} = L_\alpha y_k + b_{k+1},\ k \geq 0. \tag{2.28}$$

In this model $y_k \in E_\alpha$ denotes the pass profile on pass k, L_α is a bounded linear operator, which maps E_α into itself and $b_{k+1} \in W_\alpha$, where W_α is a linear subspace of E_α. Also, the term $L_\alpha y_k$ describes the contribution of pass k to pass $k+1$ and b_{k+1} represents inputs and other effects, which enter on the current pass.

2.4.1 Asymptotic stability

As aforementioned, the unique control problem for LRPs is that the output sequence of pass profiles $\{y_k\}_{k\geq 1}$ can contain oscillations, which can increase in amplitude in the form pass to pass direction (k). Hence a natural definition of stability is to request that bounded input sequences would produce the bounded output (pass profiles) sequences.

Definition 2.1 *[17, 18] Suppose that $||\cdot||$ denotes the norm on E_α. Then so-called asymptotic stability holds provided there exist real numbers $M_\alpha > 0$ and $\lambda_\alpha \in (0,1)$ such that $||L_\alpha^k|| \leq M_\alpha \lambda_\alpha^k,\ k \geq 0$ (where $||\cdot||$ is also used to denote the induced operator norm).*

Remark 2.5 *It is to note here that above definition concerns either discrete LRP of (2.9)-(2.10) or (2.19)-(2.20) and differential LRP of (2.14)-(2.15). It is due to the fact that asymptotic stability involves only pass-to-pass direction and in considered cases of LRPs this variable (k) is always discrete.*

If asymptotic stability holds, then the sequence of pass profiles converge in the pass-to-pass direction to a so-called limit profile, which in the case of the processes defined by (2.9)-(2.10) or (2.19)-(2.20) or the differential defined by (2.14)-(2.15) is defined by a 1D discrete or differential, respectively, linear system state-space model.

To establish conditions for asymptotic stability of the considered example therefore requires the computation of the spectral radius of the corresponding L_α. Also if asymptotic stability holds, then $\{y_k\}_{k\geq 1}$ converges strongly (in the from pass to pass direction) to the so-called steady, or limit, profile y_∞, which is the solution of the following equation

$$y_\infty = L_\alpha y_\infty + b_\infty.$$

The following theorem regards the asymptotic stability for processes described by (2.9)-(2.10) or (2.14)-(2.15) with the simplest boundary conditions.

Theorem 2.1 *The considered LRP is asymptotically stable if and only if*

$$r\big(D_0\big) < 1. \tag{2.29}$$

Then the resulting limit profile for the discrete LRP of (2.9)-(2.10) is described by the 1D linear system (with $D = 0$ for ease of presentation)

$$\begin{aligned} x_\infty(p+1) &= (A + B_0(I - D_0)^{-1}C)x_\infty(p) + Bu_\infty(p), \\ y_\infty(p) &= (I - D_0)^{-1}Cx_\infty(p), \\ x_\infty(0) &= d_\infty, \end{aligned}$$

where d_∞ denotes the strong limit of the pass state initial vector sequence $\{d_{k+1}\}_{k\geq 0}$.

For the differential LRP of (2.14)-(2.15), the resulting limit profile becomes (with $D = 0$ for ease of presentation)

$$\begin{aligned} \dot{x}_\infty(t) &= (A + B_0(I_m - D_0)^{-1}C)x_\infty(t) + Bu_\infty(t), \\ y_\infty(t) &= (I_m - D_0)^{-1}Cx_\infty(t), \\ x_\infty(0) &= d_\infty. \end{aligned}$$

Example 2.2 *To explain how the property of asymptotic stability does not guarantee that the limit profile has 'acceptable' along the pass dynamics neither for discrete nor differential LRPs, where the most basic requirement is stability in the 1D sense, i.e.* $r(A + B_0(I - D_0)^{-1}C) < 1$ *(discrete) and* $Re\big(\lambda_1(A+B_0(I-D_0)^{-1}C)\big) < 0$ *(differential) — a point which is easily illustrated by, for example, the following cases when*

- *discrete LRP -* $A = -0.5,\ B = 0, B_0 = 0.5 + b_0,\ C = 1, D = D_0 = 0$ *and the real scalar* b_0 *is chosen such that* $|b_0| \geq 1$.
- *differential LRP -* $A = -1,\ B = 0, B_0 = 1 + b_0,\ C = 1, D = D_0 = 0$ *and the real scalar* b_0 *is chosen such that* $b_0 > 0$.

Theorem 2.1 regards only LRPs of (2.9)-(2.10) and (2.14)-(2.15) with boundary conditions defined in its simplest form (2.11) – discrete case and (2.16) – differential case. It does not concern the case, when the dynamical boundary conditions [17, 59] are present. This condition cannot be applied as well to check the asymptotic stability of generalized LRP of (2.19)-(2.20). Nevertheless, the idea and the definition of asymptotic stability for those special cases hold. Hence for those cases it is very convenient to use the 1D equivalent model of LRP, defined by (2.24)-(2.25) with the appropriate entries of model matrices.

Theorem 2.2 *[17, 40, 18] Any discrete LRP given as the 1D equivalent model (2.24)-(2.25) is asymptotically stable if and only if the following holds*

$$r(\Phi) < 1. \tag{2.30}$$

For the basic LRPs of (2.9)-(2.10) the condition (2.30) is reduced to (2.29) since the system matrix Φ is lower-triangular and has repeating D_0 along its main diagonal. Hence the eigenvalues of Φ are the same as D_0 and what follows the spectral radius is the same. For the generalized discrete LRPs, Φ in general is a full matrix hence such a reduction does not hold.

2.4.2 Stability along the pass

Note that asymptotic stability guarantees the existence of a limit profile but it does not guarantee that this limit profile treated as a 1D system is stable. The reason, why it works like that is due to the fact that asymptotic stability does not concern the dynamics along the pass (along p or t). To see that this property does not guarantee that the limit profile has 'acceptable' along the pass dynamics refer to Example 2.2.

These cases, where the limit profile is unstable as a 1D linear system, are not acceptable. Hence the stronger concept of stability – stability along the pass, must be used. This stronger stability demands the BIBO property to hold independently of the dynamics, i.e. in the along the pass direction p and from pass to pass k. Introduce the formal definition of stability along the pass as follows.

Definition 2.2 *[17, 18] In terms of the abstract model of (2.28), stability along the pass holds provided there exist the real numbers $M_\infty > 0$ and $\lambda_\infty \in (0, 1)$, which are independent of α such that $||L_\alpha^k|| \leq M_\infty \lambda_\infty^k$, $k \geq 0$.*

In terms of characteristic polynomials, stability along the pass can be characterized as follows [17, 18, 20]

Theorem 2.3

- *Discrete LRP with the characteristic polynomial defined as (2.12) is stable along the pass if and only if*
$$\mathcal{C}_{\text{discreteLRP}} \neq 0 \;\; \forall (z_1, z_2): \; |z_1| \leq 1, \; |z_2| \leq 1. \tag{2.31}$$
- *Differential LRP with the characteristic polynomial defined as (2.17) is stable along the pass if and only if*
$$\mathcal{C}_{\text{diffLRP}} \neq 0 \;\; \forall (s, z): \; \text{Re}(s) \geq 0, \; |z| \leq 1. \tag{2.32}$$

The equivalent condition for Theorem 2.3 for the stability along the pass of the discrete LRP of (2.9)-(2.10) takes the following form

Theorem 2.4 *[17, 18, 20] Discrete LRP of (2.9)-(2.10) is stable along the pass, if the following hold*

- $r(D_0) < 1$,
- $r(A) < 1$,
- *all eigenvalues of the transfer function*
$$G(z) = C(zI_n - A)^{-1}B_0 + D_0,$$
$\forall \, |z| = 1$ *have modulus strictly less than unity.*

Note here that for this result $z := z_1^{-1}$ (see (2.12)).

For the differential case of (2.14)-(2.15) the following counterpart of Theorem 2.4 is presented.

Theorem 2.5 *[17, 18, 76] Differential LRP of (2.14)-(2.15) is stable along the pass, if the following hold*

- $r(D_0) < 1$,
- $Re(\lambda_i(A)) < 0$ $i = 1, \dots, n$, *where* $\lambda_i(\cdot)$ *denotes the ith eigenvalue of* $(\cdot)$,
- *all eigenvalues of the transfer function*

$$G(s) = C(sI_n - A)^{-1}B_0 + D_0,$$

  $\forall\, s = \imath\omega,\ \omega \geq 0$ *have modulus strictly less than unity.*

Note that these conditions are valid also 2D discrete linear systems described by a Roesser model of (2.1) (RM) whose stability is governed (after the appropriate modifications). This is a known fact [17] and immediately provides the interchange of stability conditions for these two classes of linear systems. Unfortunately, none of these tests provide an effective basis on which to design control laws for LRPs. In the case of LRPs (and, in particular, applications areas such as ILC) it is also essential to include performance demands into the design specification. Also, it has been found that only a limited number of key systems theoretic features (e.g. one form of controllability) for these processes can be characterized by the direct application of the theory developed already for RM (or alternatives).

Remark 2.6 *As aforementioned, there arises the question of applicability of the stability conditions presented in this section. In [41] it was presented that the conditions of Theorems 2.3, 2.4 or 2.5 belong to the class of $\mathcal{NP}$-hard problems. Hence they are hard to apply in practice or even, in some cases can be impossible to be applied. It is due to the fact that those conditions require dealing with polynomials in two variables (Theorem 2.3) and since there are no sufficient methods for dividing such polynomials, those results remain rather theoretical. On the other hand, conditions given in Theorems 2.4 and 2.5 require to check all possible complex numbers satisfying some constraints. Since there is an infinite amount of such complex numbers, it is straightforward to conclude that those conditions remain to be of the theoretical significance only, as well.*

The other significant difficulty arises in defining the synthesis (controller design towards stability along the pass) using the above conditions and this fact also limits the applicability of them.

The efficient numerically methods for analysis (testing stability along the pass) and synthesis (the controller design) of LRPs are presented in Chapter 4 of this dissertation.

2.5 Practical applications of LRPs

As aforementioned, 1D models in many cases do not match good enough the modeled physical process, hence it is purposeful to use 2D (nD) theory to the physical process modeling. Thanks to application of 2D (nD) techniques, it is possible to solve the problems of the analysis and synthesis more accurately. It is to note now that using the 1D classical theory for those cases did not work at all or did not give the required (good enough) results. Below some practical applications of physical processes, which have been found as necessary to be modeled and considered using 2D approach are presented.

2.5.1 Metal rolling

Metal rolling is a very common industrial process, where the deformation of the piece of the material takes a place. It is done with two rolls with parallel axes revolving in the opposite directions. Figure 2.3 presents a schematic diagram of the process, where the goal is to pass the material to be rolled to a pre-specified thickness through a series of rolls for successive reductions of the width. It is to note that it can be 'costly' in terms of the equipment required. A more reasonable route, from the economic point of view, is to use a single two high stand, where this process is often termed 'clogging'.

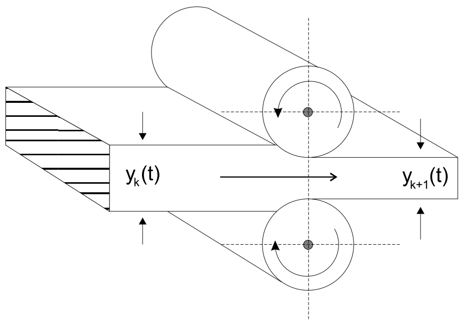


Figure 2.3. Schematic diagram of metal rolling operation

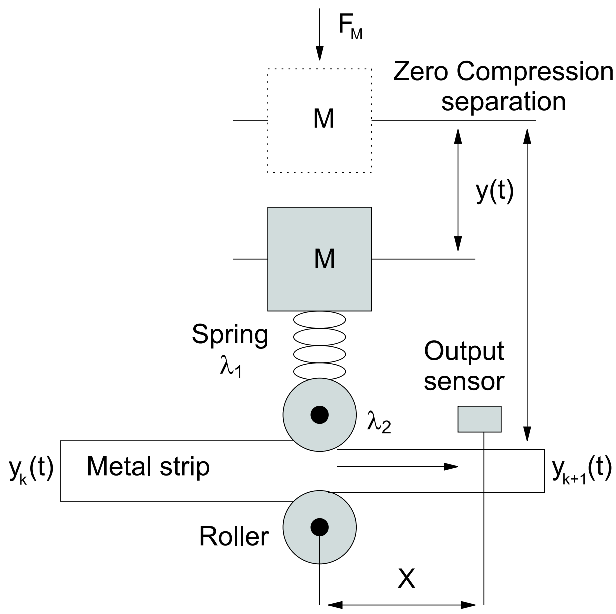


Figure 2.4. Metal rolling process

In the practical applications, many models of this process can be developed based on assumptions of the dynamics describing the various modes of operation under the consideration. Here, however, it will suffice to develop a linearized model of the dynamics of the (simplified but feasible) case shown in Figure 2.4.

The particular task, considered here, is described as the development of a simplified model relating the gauge on the current and previous passes through the rolls. These are denoted by $y_{k+1}(t)$ and $y_k(t)$ respectively and the other process variables and physical constants can be defined as follows:

F_M is the force developed by the motor,

F_s is the force developed by the spring,

M is the lumped mass of the roll-gap adjusting mechanism,

λ_1 is the stiffness of the adjustment mechanism spring,

λ_2 is the hardness of the metal strip,

$\lambda = \frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}$ is the composite stiffness of the metal strip and the roll mechanism.

To model the basic process dynamics, refer again to Figure 2.4 and following [23] or [24] first note that the force developed by the motor is

$$F_M = F_s + M\ddot{y}(t)$$

and the force developed by the spring is given by

$$F_s = \lambda_1[y(t) + y_{k+1}(t)].$$

This last force is also applied to the metal strip by the rolls and hence

$$F_s = \lambda_2[y_k(t) - y_{k+1}(t)].$$

Hence the following linear differential equation presents the relationship between $y_{k+1}(t)$ and $y_k(t)$ under the above assumptions

$$\ddot{y}_{k+1}(t) + \frac{\lambda}{M}y_{k+1}(t) = \frac{\lambda}{\lambda_1}\ddot{y}_k(t) + \frac{\lambda}{M}y_k(t) - \frac{\lambda}{M\lambda_2}F_M. \tag{2.33}$$

Suppose now that the differentiation in (2.33) is approximated by the backward difference with sampling period T. Then the resulting difference-domain approximation is

$$\begin{aligned} y_{k+1}(t) =& a_1y_{k+1}(t-T) + a_2y_{k+1}(t-2T) + a_3y_k(t) \\ &+ a_4y_k(t-T) + a_5y_k(t-2T) + bF_M, \end{aligned} \tag{2.34}$$

where

$$a_1 = \frac{2M}{\lambda T^2 + M},\ a_2 = \frac{-M}{\lambda^2 T + M},\ a_3 = \frac{\lambda}{\lambda T^2 + M}\left(T^2 + \frac{M}{\lambda_1}\right),$$

$$a_4 = \frac{-2\lambda M}{\lambda_1(\lambda T^2 + M)},\ a_5 = \frac{\lambda M}{\lambda_1(\lambda T^2 + M)},\ b = \frac{-\lambda T^2}{\lambda_2(\lambda T^2 + M)}.$$

Now set $t = pT$ and $y_{k+1}(p) = y_{k+1}(pT)$. Then (2.34) can be written as

$$\begin{aligned} x_{k+1}(p+1) &= Ax_{k+1}(p) + Bu_{k+1}(p) + B_0y_k(p), \\ y_{k+1}(p) &= Cx_{k+1}(p) + Du_{k+1}(p) + D_0y_k(p), \end{aligned}$$

where

$$x_{k+1}(p) = \begin{bmatrix} y_{k+1}(p-1) & y_{k+1}(p-2) & y_k(p-1) & y_k(p-2) \end{bmatrix}^T,\ u_{k+1}(p) = F_M,$$

and

$$A = \begin{bmatrix} a_1 & a_2 & a_4 & a_5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \; B = \begin{bmatrix} b \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; B_0 = \begin{bmatrix} a_3 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$
$$C = \begin{bmatrix} a_1 & a_2 & a_4 & a_5 \end{bmatrix}, \; D = b, \; D_0 = a_3.$$

This last state-space model becomes a special case discrete LRP of (2.9)-(2.10) and in the sequel of the dissertation the following numerical data is used $\lambda_1 = 0.6$, $\lambda_2 = 2$, $M = 0.1$ and $T = 0.1$. This set of numerical data yields $\lambda = 0.4615$ and

$$A = \begin{bmatrix} 1.9118 & -0.0047 & -1.4706 & 0.7353 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \; B = \begin{bmatrix} -0.0221 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; B_0 = \begin{bmatrix} 0.7794 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$
$$C = \begin{bmatrix} 1.9118 & -0.0047 & -1.4706 & 0.7353 \end{bmatrix}, \; D = \begin{bmatrix} -0.0221 \end{bmatrix}, \; D_0 = \begin{bmatrix} 0.7794 \end{bmatrix}. \tag{2.35}$$

2.5.2 Long-wall coal cutting

The coal mining processes are the first of the historical significance industrial processes, which have been modeled by the LRPs. It has been turned out that the control schemes based upon LRPs provide satisfactory results over other applied alternatives.

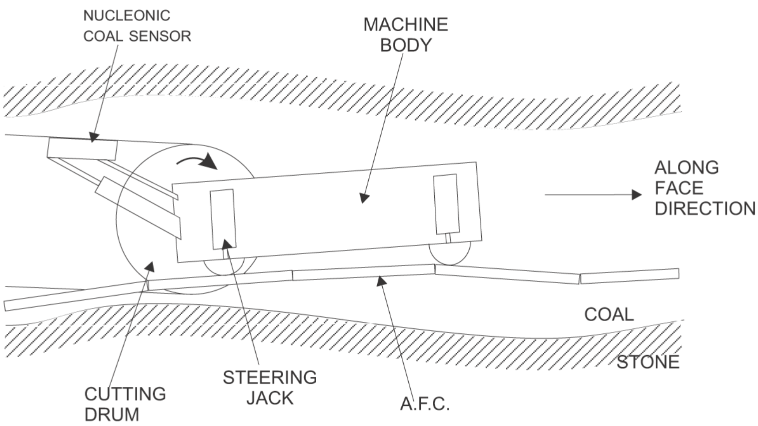


Figure 2.5. Long-wall coal cutting process

Figures 2.5 and 2.6 illustrate the operation of the long-wall coal cutting process, in which the coal cutting machine is moved along the entire length of the coal face. The coal is cut with the rotating cutting drum. After reaching the end of the coal face, the machine is moved back at the high speed to the beginning of the face and then shifted to the face by the width of the already cut coal. Then the whole process is started again. The gained coal is transported away by the Armoured Face Conveyor (denoted A.F.C.). The described simplest mode of that operation expects the cutting of the coal in only one direction, however it is also possible to provide the schedule to cut the coal in both directions, i.e. after reaching the end of the coal face

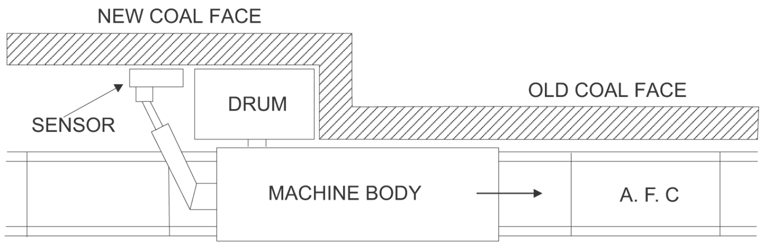


Figure 2.6. Long-wall coal cutting process

the machine is not moved back but the drum is shifted to the face of the coal and the process of cutting is started again in the backward direction.

It is to note that due to the current requirements, the cutting drum can be moved in the vertical direction (raised or lowered) with respect to the A.F.C. To provide the basic measures a nucleonic coal sensor has been mounted at the beginning of the machine.

The mathematical modeling, which allows obtaining the appropriate model of the structure of LRPs for this physical example has been presented in details in [17, 20].

2.5.3 Iterative Learning Control

Application of Iterative Learning Control (ILC) has been motivated by the necessity of improving the track (decreasing the tracking error) that considered system tried to follow repeatedly. During recent years the applications, concepts and problems arising in the area of ILC have been widely investigated [26, 34, 36, 77, 29, 78, 79]. ILC concept relies on repeatedly decreasing the tracking error during the sequence of iterations (trials, sweeps, repetitions or, in terms of LRPs, passes). The original idea of ILC arises from observations made to the human ways of learning by repetitions. It is used mainly in learning of artificial intelligence systems (e.g. neural networks or robots).

Due to the inherent 2D structure of ILC (information propagate in two independent directions, i.e. dynamics of the system itself during the single iteration - time; and the iteration count), it is natural to apply the 2D techniques for the control purposes. Those include the stability investigation (analysis problem) and/or controller design towards stability in the closed loop configuration (synthesis problem). There also arises the problem of ensuring the required performance (e.g. quick convergence) during the learning procedure.

It is to underline that the concept of ILC relies on using the information come from the former (or formers) trial to tune (control) the behaviour of the current one to attain the required performance (i.e. to minimize the tracking error).

To describe the process of ILC in 2D terms, let a single learning iteration (trial) numbered by k be denoted by the execution of the discrete system defined as

$$\begin{aligned} x_k(p+1) &= Ax_k(p) + Bu_k(p), \\ y_k(p) &= Cx_k(p), \end{aligned} \tag{2.36}$$

where $u_k(p) \in \mathrm{R}^r$ - the input vector, $y_k(p) \in \mathrm{R}^m$ - the output vector (the current trajectory), $x_k(p) \in \mathrm{R}^n$ - the state vector and the desired output trajectory (denoting the required to attain track) $y_{ref}(p)$ and where $0 \leq p \leq \alpha-1$. The boundary conditions become

$$\begin{aligned} x_k(0) &= x_0, \quad k = 0, 1, \dots, \\ u_0(p) &= 0, \quad p = 0, 1, \dots, \alpha. \end{aligned} \tag{2.37}$$

Using the above notation the problem formulated as ILC becomes: through several iteration find (learn) the appropriate control sequence $u_k(p)$, $0 \leq p \leq \alpha-1$ such that the tracking error, i.e. the difference between the current trajectory at iteration k and the desired output trajectory converges to 0 along the full execution (i.e. $0 \leq p \leq \alpha-1$). To formalize this, the tracking error is defined by

$$e_k(p) = y_{ref}(p) - y_k(p). \tag{2.38}$$

The minimization of $e_k(p)$ $0 \leq p \leq \alpha-1$ is done by adjusting the input from the current iteration i.e. $u_k(p)$ to a new input $u_{k+1}(p)$ for the next iteration with some correction factor. Therefore, a general iterative control sequence is defined as

$$u_{k+1}(p) = u_k(p) + \Delta u_{k+1}(p), \tag{2.39}$$

where $\Delta u_k(p)$ denotes the correction of the control input required to improve the control (to minimize the tracking error). The ILC scheme is called to be convergent if $e_k(p) \to 0, \ \ k \to \infty$, $0 \leq p \leq \alpha-1$. In that case the pass profile $y_k(p)$ tends to the reference signal $y_{ref}(p)$ for each $0 \leq p \leq \alpha-1$. In what follows, combining (2.36), (2.38) and (2.39) provides the following dependence

$$e_{k+1}(p) - e_k(p) = -CA\eta_{k+1}(p) - CB\Delta u_k(p-1),$$

where

$$\eta_{k+1}(p) \triangleq x_{k+1}(p-1) - x_k(p-1).$$

What is more, from (2.39) and (2.36)

$$\eta_k(p+1) = A\eta_k(p) + B\Delta u_k(p-1).$$

Hence the correction of the control law is given as

$$\Delta u_{k+1}(p) = K_1\eta_{k+1}(p+1) + K_2 e_k(p+1)$$

and the LRP, which models the ILC scheme is obtained

$$\begin{bmatrix} \eta_{k+1}(p+1) \\ e_{k+1}(p) \end{bmatrix} = \begin{bmatrix} A - BK_1 & -BK_2 \\ -CA + CBK_1 & I - CBK_2 \end{bmatrix} \begin{bmatrix} \eta_{k+1}(p) \\ e_k(p) \end{bmatrix} \tag{2.40}$$

It is clear [36, 33] that asymptotic stability of the LRP given in the form of (2.40) guarantees that the underlying ILC law (2.39) is convergent and hence the explicit link between the efficient numerical computation task and the system theoretic approach exists. Along the same lines, the stability along the pass requirement prevents the presence of the undesirable dynamics in the along the pass direction (p) and is crucial especially, when passes are long.

To complete the description, it is necessary to provide the boundary conditions as

$$\begin{aligned} \eta_k(0) =& x_{k+1}(0) - x_k(0) = x_0 - x_0 = 0, \quad k = 0, 1, \dots, \\ e_0(p) =& y_{ref}(p) - y_0(p) = y_{ref}(p) - CA^T x_0, \quad p = 0, 1, \dots, \alpha-1. \end{aligned}$$

Note that here only the discrete case of the basic one-iteration system has been considered. It is natural that the analogous continuous dynamical system describing the single execution of the ILC process can be assumed. In such a case the resulting 2D model has a discrete-continuous structure. Such a continuous case of ILC can be found e.g. in [50].

Remark 2.7 *Note that, when considering ILC due to the fact that the variable p associated with the time during the iteration (the time duration) is finite by the definition, it is quite natural to apply the LRP model and to employ all control techniques known for LRPs. The other important point is to underline the purposefulness of application of LRPs over the alternative 2D models arises, when the continuous case of ILC (i.e. the basic system (2.36) is continuous instead of discrete) is considered. There is no basic 2D models involving the discrete-differential couple of variables. Nevertheless, LRPs are considered to occur as double discrete variables model (see (2.9)-(2.10)) or as discrete-differential variables model (see (2.14)-(2.15)).*

It is quite natural, to note that representation of the ILC in the valid LRP form, provides some possibilities. Namely, note that the stability of the LRP in this case is indeed related with the convergence of the ILC process which is considered. Analogously, the controller design problem of LRP in terms of ILC can be treated as a problem of ensuring the convergence of the iterative procedure. Hence the links between the systems theory and the features of the data processing (which clearly includes the ILC schemes) are stated.

It is clear that the ILC scheme can be efficiently extended to modeling and next, improving, other iterative computational tasks.

2.5.4 Spatially interconnected systems

Recently, there appeared the interest in so-called spatially interconnected systems (see e.g. [32] and references therein), which clearly have strong links to LRPs. Such models can successfully model the spatially complex systems composed of the basic subsystems shown in Figure 2.7).

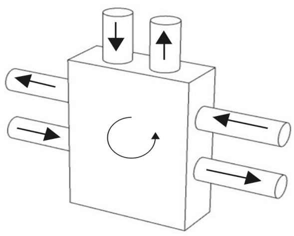


Figure 2.7. The basic idea of the subsystem

The reported in the literature applications include: automated highway systems, air–forma tion flight, satellite constellations, cross directional control in paper processing applications and micro-cantilever array control for massively parallel data storage. Note that depending on the way in which the subsystems are connected, it is possible to obtain several different structures of the spatially interconnected systems. The simplest structures are presented in Figures 2.8 and 2.9 (for the others refer to [32]). It is to note that those structures are strongly depended on the nature of the considered system.

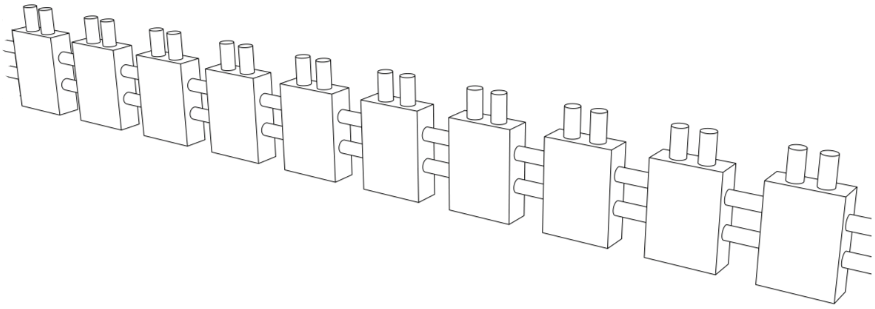


Figure 2.8. The possible interconnection of the subsystems - the infinite line

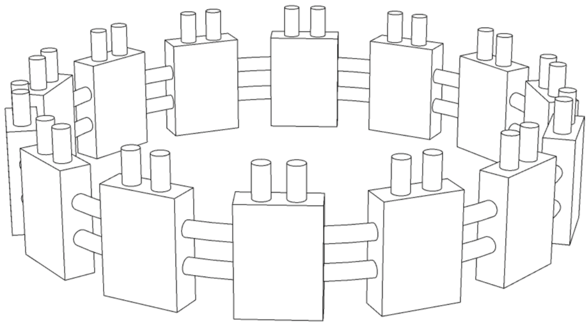


Figure 2.9. The possible interconnection of the subsystems - the circle

It is to underline that the state-space models presented in [32] for the spatially interconnected systems are (after the appropriate reformulations) similar to the models of LRPs. Hence the application of the analysis/synthesis methods developed originally for LRPs is supposed to be fruitful for the both areas.

Chapter 3

Linear Matrix Inequalities in control

Most of the control problems considered in this dissertation can be presented in the forms of a few convex optimization problems involving Linear Matrix Inequalities (LMIs) (see [42, 43, 44, 45, 80, 81]). Since the numerically powerful interior-point methods algorithms have been implemented recently, the main task at the moment is to present the considered problem in the framework of known, solvable by those algorithms, scheme. Nevertheless, it is worth of noting that such a transformation into the LMI in many cases (e.g. the robust stability condition) is not a straightforward trivial operation and requires involving the sophisticated methods of mathematical proving.

The wide list of the control problems, which can be presented and finally solved with LMI, can be found in the [42], however this dissertation's topics are due to the Lyapunov theory for the linear 2D systems. For the classical 1D systems, the application of LMI provides the natural reformulation of the stability/controller design conditions and the results still remains the necessary and sufficient conditions. For 2D systems and LRPs there has been proved that some of analysis/synthesis conditions, however necessary and sufficient, are $\mathcal{NP}$-hard. Hence those conditions are considered to be not applicable in practice. On the other hand, it has been presented that the application of LMIs to analysis and synthesis of 2D systems and LRPs provides the $\mathcal{P}$-class conditions. Note however that those conditions are only sufficient now. Therefore, from the point of view of numerical effectiveness, LMI methods alone can be treated as very efficient algorithmic tools for solving complicated control related computational problems.

3.1 Basics of LMIs

Linear Matrix Inequality is defined as any constraint of the following form:

$$F(x) := F_0 + x_1 F_1 + x_2 F_2 + \ldots + x_M F_M < 0, \tag{3.1}$$

where

$x = (x_1,\ x_2, \ldots, x_M)$ - a vector of unknown scalars (the *decision* or *optimization* variables),

$F_0,\ F_1, \ldots, F_M$ - given symmetric matrices,

< 0 stands for "negative definite", i.e. for any nonzero vector $u \in \mathrm{R}^M$, $uF(x)u^T < 0$ (the largest eigenvalue of $F(x)$ is negative).

The constrains $F(x) > 0$, $F(x) < G(x)$ and $G(x) > 0$, $F(x) > 0$ are the special cases of (3.1), since they can be rewritten as $-F(x) < 0$, $F(x) - G(x) < 0$ and $\mathrm{diag}(G(x), F(x)) > 0$, respectively.

The LMI (3.1) is a convex constraint on x, since for any y, z satisfying $F(y) < 0$ and $F(z) < 0$, imply that $F(\frac{z+y}{2}) < 0$.

If the solution set of (3.1), called the feasible set, exists, then it is a convex subset of R^M. Note that finding a solution x satisfying (3.1), can be presented as the convex optimization problem.

In some cases, the strict inequality in (3.1) is replaced by the non-strict one. In such a case (3.1) becomes

$$F(x) := F_0 + x_1F_1 + x_2F_2 + \ldots + x_MF_M \leq 0 \tag{3.2}$$

and all of the above features are satisfied as well.

The inequality of (3.1) is called the canonical form of LMI since the sought solution for given LMI is a vector of decision variables x_i, $i = 1, 2, \ldots, M$. In the sequel, LMIs in the canonical form are seldom used, since the considered problems are due to finding a variable matrix (say X or P) such that the appropriate inequality holds. It is easy to show that any "standard" LMI, involving finding the matrix variable can be presented in the canonical form.

Example 3.1 *Consider the Lyapunov stability condition of differential 1D system described by*

$$\dot{x}(t) = Ax(t), \tag{3.3}$$

which states that the system is stable if and only if there exists a positive definite matrix $P > 0$ such that the following LMI is satisfied

$$A^TP + PA < 0, \quad P = P^T > 0 \tag{3.4}$$

or

$$\begin{bmatrix} A^TP + PA & 0 \\ 0 & -P \end{bmatrix} < 0.$$

To present this condition in the canonical form of LMI note that

$$A = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix}, \; P = P^T = \begin{bmatrix} p_{11} & \ldots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{1n} & \ldots & p_{nn} \end{bmatrix}$$

For the ease of the presentation and without the loss of the generalization assume that $n = 2$. Next, the multiplication of (3.4), due to the structures of matrices A and P, provides the following inequality (note that since a_{ij} and p_{kl} are scalars then $a_{ij}p_{kl} = p_{kl}a_{ij}$)

$$\begin{bmatrix} 2p_{11}a_{11} + 2p_{12}a_{21} & p_{12}(a_{11}+a_{22}) + p_{22}a_{21} + p_{11}a_{12} & 0 & 0 \\ p_{12}(a_{11}+a_{22}) + p_{22}a_{21} + p_{11}a_{12} & 2p_{12}a_{12} + 2p_{22}a_{22} & 0 & 0 \\ 0 & 0 & -p_{11} & -p_{12} \\ 0 & 0 & -p_{12} & -p_{22} \end{bmatrix} < 0$$

or write it in the canonical form of LMI as

$$p_{11}\begin{bmatrix} 2a_{11} & a_{12} & 0 & 0 \\ a_{12} & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + p_{12}\begin{bmatrix} 2a_{21} & a_{22}+a_{11} & 0 & 0 \\ a_{22}+a_{11} & 2a_{12} & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} + p_{22}\begin{bmatrix} 0 & a_{21} & 0 & 0 \\ a_{21} & 2a_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} < 0.$$

To conclude the above example, it has been presented that the LMI with the matrix variable P has been transformed into its canonical form. Note that since LMIs are symmetric, it is possible to transform any other LMI using the same manner. Hence it is straightforward to see that any LMI with the matrix variables can be presented in the appropriate canonical form.

3.2 Bilinear Matrix Inequalities

However, the problems considered in this dissertation are treated finally to become LMIs, most of those problems are not given in the LMI form directly. One way to deal with such a problems is to try to linearize them to obtain LMI. For instance, consider the 1D differential system synthesis problem presented in [81], where the task is to design the controller K ensuring that the closed loop system is stable. The inequality corresponding to that problem can be written as

$$(A+BK)^T P + P(A+BK) < 0, \quad P > 0 \tag{3.5}$$

or, after multiplication

$$XA^T + AX + K^T B^T P + PBK < 0, \quad P > 0.$$

This is not an LMI since there is a multiplication between the variable matrices K and P. To convert this problem into the LMI form, left- and right- multiply it by $X = P^{-1}$ and introduce the following substitution $K = NX^{-1}$. Hence the inequality (3.5) becomes the following LMI

$$XA^T + AX + N^T B^T + BN < 0, \quad X > 0.$$

It is to note that such a conversion to LMI, in general is not possible to be performed for all problems. Hence recently, there has arisen another opportunity to deal with problems in form similar to (3.5) (e.g. involving two decision variables multiplication). Thus, Bilinear Matrix Inequalities (BMIs) have been defined as a general way to handle such a problems [82]

$$F(x,y) = F_0 + \sum_{i=1}^{M_1} x_i F_i + \sum_{j=1}^{M_2} y_j G_j + \sum_{i=1}^{M_1}\sum_{j=1}^{M_2} x_i y_j H_{ij} \geq 0, \tag{3.6}$$

where $x \in \mathrm{R}^{M_1}$, $y \in \mathrm{R}^{M_2}$ are the decision variables vectors, F_0, F_i, $1 \leq i \leq M_1$, G_j, $1 \leq j \leq M_2$ and H_{ij}, $1 \leq i \leq M_1$, $1 \leq j \leq M_2$ are given symmetric matrices.

As it was mentioned, in general, BMIs are non-convex optimization problems, which can have multiple local solutions, hence solving a general BMI was shown to be $\mathcal{NP}$-hard [83]. On the other hand, in many cases, no such a simple recasting method to the LMI as for (3.5) can be presented.

Today, problems given in the form of BMIs can be solved with algorithms based on a spatial branch and bound strategy but unfortunately they are suitable only for small-size problems. Due to the fact that there remain some open problems regarding the performance and the implementation of BMIs solvers, these methods are still being developed [84, 85, 86].

The principal method for reformulating problems given as BMIs into the terms of LMIs is the so-called Schur complement accompanied by the suitable transformations (see, e.g. [50]).

3.3 Schur complement

As aforementioned, for the problems, which are not originally LMIs, it is necessary to try to convert those into LMIs. Very useful in this field appears to be so-called Schur complement [42, 87, 88], which allows to disconnect the multiplied variables.

Lemma 3.1 *(Schur Complement [42, 87, 88]) Let Σ_1, Σ_2, Σ_3 and Σ_4 be real matrices of appropriate dimensions. Then for $\Sigma_1 > 0$ and $\Sigma_3 = \Sigma_3^T$*

$$\Sigma_3 + \Sigma_2^T \Sigma_1 \Sigma_2 < 0$$

if and only if

$$\begin{bmatrix} \Sigma_3 & \Sigma_2^T \\ \Sigma_2 & -\Sigma_1^{-1} \end{bmatrix} < 0 \ or \ \begin{bmatrix} -\Sigma_1^{-1} & \Sigma_2 \\ \Sigma_2^T & \Sigma_3 \end{bmatrix} < 0.$$

3.4 Geometry of LMIs

The set of feasible solutions of the considered LMI (3.1) (the feasibility set) is denoted as follows

$$\mathcal{F}(x) = \big\{ x \in \mathrm{R}^M : F(x) = F_0 + \sum_{i=1}^{M} x_i F_i < 0 \big\}. \tag{3.7}$$

Due to the fact that LMI is defined in the space of its decision variables ($x \in \mathrm{R}^M$), it is possible to present the feasibility set as a geometrical shape in this space. Note that (3.1) or (3.2), due to matrices F_i denotes the convex subspace of R^M. The feasibility region of the considered strict LMI is the interior without boundaries of that subspace (for the non-strict LMI the feasibility region is the same interior but with boundaries included). For the positive (non-negative) definiteness of $F(x)$, it is required that all of its diagonal minors to be positive (non-negative). For the negative (non-positive) definiteness of $F(x)$, it is required that its diagonal minors of odd degree to be negative (non-positive) and the minors of even degree to be positive (non-negative), respectively.

It is straightforward to see that the diagonal minors are multi-variate polynomials of variables x_i. Hence the LMI set can be described as (φ denotes the size of the considered LMI)

$$\mathcal{F}(x) = \big\{ x \in \mathrm{R}^M : f_i(x) > 0, \ i = 1, \dots, \varphi \big\} \tag{3.8}$$

for the positive definite LMI and the following for the negative one

$$\mathcal{F}(x) = x \in \mathrm{R}^M : \begin{cases} f_i(x) < 0, & \forall \text{ odd } i : \ 1 \le i \le \varphi, \\ f_i(x) > 0, & \forall \text{ even } i : \ 2 \le i \le \varphi, \end{cases} \tag{3.9}$$

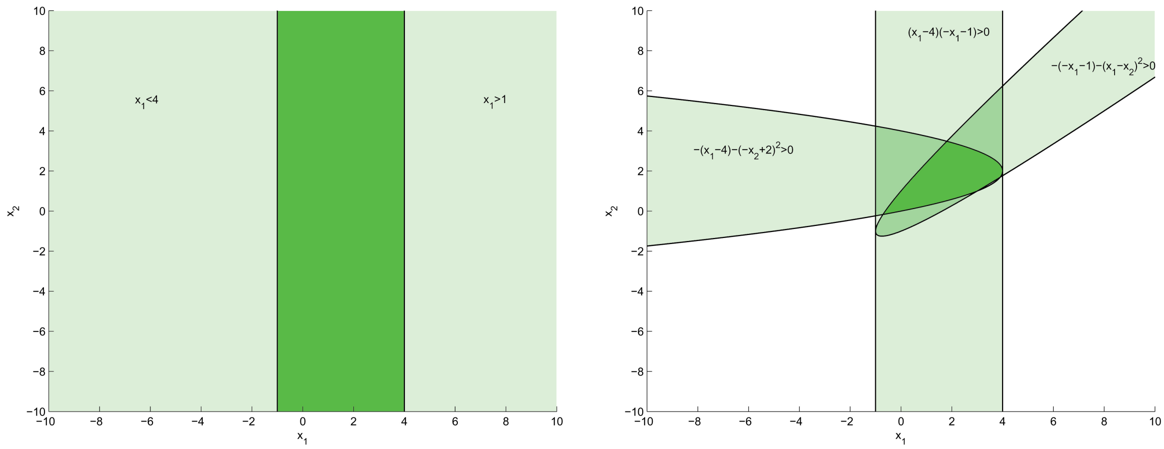
which are the semi-algebraic sets. Moreover, they are the convex sets.

Example 3.2 *To see the result of the previous paragraph, consider the following LMI*

$$F(x_1, x_2) = \begin{bmatrix} x_1 - 4 & -x_2 + 2 & 0 \\ -x_2 + 2 & -1 & x_1 - x_2 \\ 0 & x_1 - x_2 & -x_1 - 1 \end{bmatrix} < 0.$$

To find a feasibility region of this LMI, write the conditions for the diagonal minors of degree: first, second and third in variables $x_1,\ x_2$. Hence the minors become

$$
\begin{cases}
\begin{array}{ll}
x_1 - 4 < 0, & \\
-1 < 0, & \textit{(first degree minors - must be negative)} \\
-x_1 - 1 < 0, & \\
\hline
-(x_1 - 4) - (-x_2 + 2)^2 > 0, & \\
-(-x_1 - 1) - (x_1 - x_2)^2 > 0, & \textit{(second degree minors - must be positive)} \\
(x_1 - 4)(-x_1 - 1) > 0, & \\
\hline
-(x_1 - 4)(-x_1 - 1) - (-x_2 + 2)^2(-x_1 - 1) & \textit{(third degree minor (the determinant of } F(x)\textit{) -} \\
-(x_1 - 4)(x_1 - x_2)^2 < 0. & \textit{- must be negative)}
\end{array}
\end{cases}
$$



a) b)

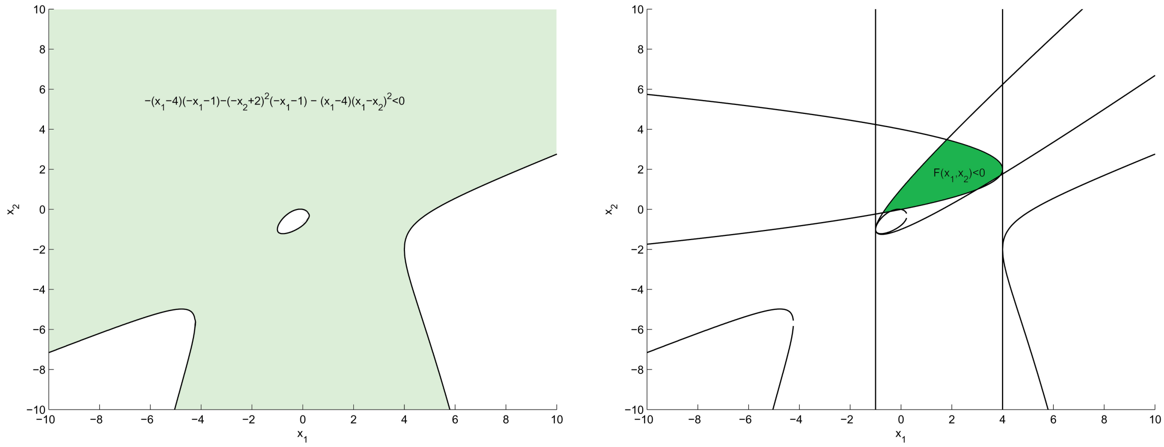
Figure 3.1. The solutions for the first (a) and second (b) degree minors

Given figures illustrate the solutions for the minors of first (Figure 3.1 a)), second (Figure 3.1 b)) and third degree (Figure 3.2 a)), respectively. Figure 3.2 b) shows the feasibility region for the considered LMI.

It is straightforward to see that the feasibility region is the intersection of the regions, which satisfy the constrains due to the corresponding minors.

The solution of the above problem given by the Matlab LMI Control Toolbox *is* $x_1 = 1.667$ *and* $x_2 = 1.833$*. Note that the point of those coordinates lays inside the feasibility region shown in Figure 3.2 b).*

From this example, it is straightforward to see that any set of LMIs $F_1(x) < 0,\ F_2(x) < 0$ can be treated and solved as $F(x) = \operatorname{diag}\big(F_1(x), F_2(x)\big) < 0$. The same fact holds for the set of LMIs dependent on distinct variables i.e. $F_1(x) < 0,\ F_2(y) < 0$, which can be presented as $F(x,y) = \operatorname{diag}\big(F_1(x), F_2(y)\big) < 0$.



a) b)

Figure 3.2. The solutions for the third (a) degree minors and the feasibility region for the considered LMI (b)

3.5 Reformulation of the LMI problems into SDPs

Since the term 'LMI' is narrowly used mainly in the control applications, for the others engineering science areas the Semidefinite Programming (SDP) is used as one, which has the similar meaning. There have been developed several of algorithms and implemented some software packages that allow to solve problems given as SDP. SDP problem is defined as follows (see [89])

$$\text{SDP}\begin{cases}\mathcal{P}:\ \text{minimize} & \sum_{i=1}^{\hat{M}} c_i x_i,\\ \ \text{subject to} & \mathbf{X}=\sum_{i=1}^{\hat{M}} G_i x_i - G_0,\\ & \mathbf{X}\geq 0,\\ \mathcal{D}:\ \text{maximize} & G_0\bullet\mathbf{Y},\\ \ \text{subject to} & G_i\bullet\mathbf{Y}=c_i\ \ (1\leq i\leq M),\\ & \mathbf{Y}\geq 0,\end{cases} \tag{3.10}$$

where $\mathbf{X}=\mathbf{X}^T,\ \mathbf{Y}=\mathbf{Y}^T\ \in \mathrm{R}^{n\times n}$- the variable matrices, $G_i\in\mathrm{R}^{n\times n}$ $i=0,1,\ldots,\hat{M}$ - constraints matrices, $c\in\mathrm{R}^{\hat{M}}$ - the cost vector, $x\in\mathrm{R}^{\hat{M}}$ - the optimization variables vector, $\bullet$ - the inner product of two matrices, ≥ 0 - semipositive definiteness. By $\mathcal{P}$ the primal form of the optimization problem is denoted and by $\mathcal{D}$ the dual form, respectively.

The standard problems which are formulated as SDP includes:

- Max-Cut Problem [90, 91],
- Graph Bisection [92],
- Maximum cliques in graphs [93],
- Min-Max Eigenvalue Problem [94, 95].
- and others.

In some cases, the following, equivalent to (3.10), definition of SDP is used

$$\text{SDP'}\begin{cases}\mathcal{P}: \text{ minimize} & \widehat{G}_0 \bullet \widehat{\mathbf{Y}}, \\ \quad \text{subject to} & \widehat{G}_i \bullet \widehat{\mathbf{Y}} = \widehat{c}_i \;\; (1 \le i \le \hat{M}), \\ & \widehat{\mathbf{Y}} \ge 0, \\ \mathcal{D}: \text{ maximize} & \sum_{i=1}^{\hat{M}} \widehat{c}_i \widehat{x}_i, \\ \quad \text{subject to} & \sum_{i=1}^{\hat{M}} \widehat{G}_i \widehat{x}_i + \widehat{\mathbf{X}} = \widehat{G}_0, \\ & \widehat{\mathbf{X}} \ge 0. \end{cases} \tag{3.11}$$

To convert SDP' of (3.11) into the original SDP of (3.10) follow the following substitutions

$$\begin{aligned} \widehat{G}_i &\rightleftarrows -G_i, \quad (0 \le i \le \hat{M}), \\ \widehat{c}_i &\rightleftarrows -c_i, \quad (1 \le i \le \hat{M}), \\ \widehat{\mathbf{Y}} &\rightleftarrows \mathbf{Y}, \\ \widehat{x}_i &\rightleftarrows x_i, \quad (1 \le i \le \hat{M}), \\ \widehat{\mathbf{X}} &\rightleftarrows \mathbf{X}. \end{aligned}$$

There is also the possibility for redefining the problem originally given as LMI into the form of SDP and this fact is of the crucial importance, when considering the application of the SDP software in duty of solving the control problems given as LMIs. Hence to redefine LMI into the valid form of SDP recall a nonnegative case of LMI defined in (3.1), given here as

$$F(x) := F_0 + x_1 F_1 + x_2 F_2 + \ldots + x_M F_M \ge 0$$

and remember that in this case the task was to find any decision variables vector x satisfying the LMI or to prove that such a vector does not exist (feasibility problem). Hence introduce the additional variable x_{M+1} to LMI and write it as a primal form of SDP

$$\begin{aligned} &\text{minimize } x_{M+1}, \\ &\text{subject to } \mathbf{X} = F_0 + \sum_{i=1}^{M} F_i x_i + x_{M+1} I, \\ &\mathbf{X} \ge 0. \end{aligned}$$

Hence it is possible to treat LMI as a special case of SDP given in the primal form, when

$$\hat{M} = M + 1,\ G_0 = -F_0,\ G_i = F_i,\ c_i = 0,\ (1 \le i \le M),\ c_{\hat{M}} = 1,\ G_{M+1} = I.$$

The LMI formulated in the form of SDP is feasible, when there exists $x_{M+1} < 0$. Note that since in the control problems, it is required that the inequality sign is negative (not non-positive), it is to see that this LMI becomes the strict one, since x_{M+1} is to be strictly negative.

Concluding, it is straightforward to notice that any LMI problem can be tried to be solved using not only LMI software but as well the SDP software. This extends the whole area of possible to apply software packages, including the parallel SDP software packages e.g. DSDP ([96]) or SDPARA ([97, 98]).

Example 3.3 *To provide the view on the procedure described previously, consider again Example 3.1. The resulting LMI has the following canonical form*

$$F(P) = p_{11}\begin{bmatrix} 2a_{11} & a_{12} \\ a_{12} & 0 \end{bmatrix} + p_{12}\begin{bmatrix} 2a_{21} & a_{22}+a_{11} \\ a_{22}+a_{11} & 2a_{12} \end{bmatrix} + p_{22}\begin{bmatrix} 0 & a_{21} \\ a_{21} & 2a_{22} \end{bmatrix} < 0$$

and it is to remember that

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = p_{11}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + p_{12}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + p_{22}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} > 0.$$

Now, it is necessary to combine $F(P) < 0$ and $P > 0$ as

$$\begin{bmatrix} -F(P) & 0 \\ 0 & P \end{bmatrix} > 0.$$

Introduce the additional decision variable p_* (x_4) and formulate this problem as primal form of SDP

$$\text{minimize } \underbrace{p_*}_{x_4},$$

$$\text{subject to}$$

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{G_0} + \underbrace{p_{11}}_{x_1}\underbrace{\begin{bmatrix} -2a_{11} & -a_{12} & 0 & 0 \\ -a_{12} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{G_1} + \underbrace{p_{12}}_{x_2}\underbrace{\begin{bmatrix} -2a_{21} & -a_{22}-a_{11} & 0 & 0 \\ -a_{22}-a_{11} & -2a_{12} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{G_2}$$

$$+ \underbrace{p_{22}}_{x_3}\underbrace{\begin{bmatrix} 0 & -a_{21} & 0 & 0 \\ -a_{21} & -2a_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{G_3} + \underbrace{p_*}_{x_4}\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{G_4} \geq 0.$$

Note that since there are no constraints on p_* (x_4), the positive inequality sign can be changed into the nonnegative one, to fulfill the requirements of SDP definition. Now, if the optimization process stops at $p_* < 0$ ($x_4 < 0$), the certificate of feasibility is given.

Example 3.4 *To highlight the above procedure the solution of the feasibility problem presented in Example 3.2 has been considered again. Now, the assumed task is to present it in the form of the SDP problem and solve it using the SDP solver. The resulting SDP problem can be presented as follows*

$$\min\ 0\,x_1 + 0\,x_2 + 1\,x_3,$$

$$\text{subject to } \begin{bmatrix} -4 & 2 & 0 \\ 2 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} + x_1\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 1 \end{bmatrix} + x_2\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + x_3\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \geq 0.$$

The application of SDPARA *solver provides the following solutions $x_1 = 2.0$, $x_2 = 2.0$ and $x_3 = -1.0$. It is easy to note that the point (x_1, x_2) lays inside the feasibility region shown in Figure 3.2 b).*

3.6 Elimination of the equality constraints

In some applications the linear equality constraints together with inequalities appear. Hence it is necessary to eliminate those constraints, since the definition of LMI or SDP does not allow such case.

As presented, the LMI condition can be presented in the canonical form of (3.1). In this case, LMI with the additional factors (the linear equality constraints) can be presented as

$$\begin{aligned} F(x) < 0,\ x \in \mathrm{R}^M, \\ Hx = b. \end{aligned}$$

Due to the fact that the equality constraints do not involve (on average) all entries of x, matrix H is not the square, full rank matrix. Hence it is possible to introduce the new vector of decision variables $y \in \mathrm{R}^{\widetilde{M}}$, which indeed is the mapped x such that as

$$y = H^{\sharp}x + y_0,$$

where $H^{\sharp}$ denotes the nullspace of H and y_0 is any particular solution of $Hx = b$. In this case the reformulated LMI becomes

$$F(y) = F(H^{\sharp}x + b) < 0$$

and the equality constraints have been eliminated.

Hence the LMI or SDP solver now can be used to solve $F(y) < 0$. However, it is to note that after obtaining the feasible solution vector y, it is necessary to re-map it back to the original variables vector x.

Example 3.5 *To present the idea of the elimination of the equality constraints consider again the analysis problem given in Example 3.1. However here, the additional constraints on P are assumed*

$$\begin{aligned} A^T P + PA < 0,\ \ P = P^T > 0, \\ \mathit{trace}(P) = 1, \end{aligned}$$

where $A = \mathit{diag}(-1, -1)$. Hence $P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$ and $p_1 + p_3 = 1$. Then the equality constraint is given as

$$p_1 + p_3 = 1 \quad \leftrightarrow \quad H\widetilde{p} = b,$$

$$\text{where } H = [\, 1\ \ 1\ \ 0\,],\ \ \widetilde{p} = [\, p_1\ \ p_3\ \ p_2\,]^T \ \text{ and } b = 1.$$

Introduce new vector of variables $\widetilde{q}$ and note that

$$\widetilde{q} = H^{\sharp}\widetilde{p} + \widetilde{p}_0,$$

where $\widetilde{p}_0$ – any particular solution to $H\widetilde{p} = 1$ e.g. $\widetilde{p}_0 = H^T(HH^T)^{-1}b$ and $H^{\sharp}$ – nullspace of H (in Matlab *can be computed by* null(H,'r')*). In this particular case, the following mapping is obtained*

$$\underbrace{\begin{bmatrix} p_1 \\ p_3 \\ p_2 \end{bmatrix}}_{\widetilde{p}} = \underbrace{\begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}}_{\widetilde{p}_0} + \underbrace{\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{H^{\sharp}} \underbrace{\begin{bmatrix} q_1 \\ q_2 \end{bmatrix}}_{\widetilde{q}}.$$

Hence under the constraint, the following change of variables is performed

$$P > 0 \;\leftrightarrow\; \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} + q_1 \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} + q_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} > 0.$$

Now, this mapping has to be substituted into the original LMI, which becomes

$$A^T\Big(\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} + q_1 \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} + q_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\Big) + \Big(\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} + q_1 \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} + q_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\Big)A < 0,$$

what, after multiplication, can be rewritten as

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} + q_1 \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} + q_2 \begin{bmatrix} 0 & -2 \\ -2 & 0 \end{bmatrix} < 0,$$

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} + q_1 \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} + q_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} > 0.$$

It is straightforward to see that there are many feasible pairs q_1 *and* q_2 *satisfying the above inequalities. Take* $q_1 = 0.3$ *and* $q_2 = 0.1$*. To obtain the final solution take* $p_1 = 0.2$*,* $p_2 = 0.1$ *and* $p_3 = 0.8$*.*

3.7 Efficient algorithms for solving LMIs/SDPs

The first step towards the effective algorithms to solve LMIs was the developing the simplex algorithm. However, it was recognized as a badly-behavioral - in the worst case, the time (total number of iterations) required to finish the optimization process, might increase exponentially in the number of the optimization variables. Hence due to that drawback, the polynomial-time algorithms were sought.

The first one, called the ellipsoid algorithm, was developed by Khachiyan ([99]). It is the simplest polynomial-time algorithm that can be applied to solve the convex optimization problem (and hence LMI). The idea of the ellipsoid algorithm is as follows: start with the ellipsoid (defined in the variables space), which contains the optimal solution. Then the procedure relying on the repetition the following sequence until the required accuracy is not reached: divide the current ellipsoid in half, choose the half in which the optimum lays, compute the minimum volume ellipsoid that contains the chosen half and finally check the stop condition (e.g. the accuracy or the iteration number). The above procedure guarantees that the optimum lays still in the inside of each ellipsoid and since the volume of each successive ellipsoid is decreased geometrically, the solution should be found quickly. Nevertheless, tests proved that in practice the optimal solution is found relatively slowly. Hence those kind of algorithms are not widely used. The details of the ellipsoid algorithm can be found in [42].

Despite the slow convergence in practical applications, when the advantages of the ellipsoid algorithm have been realized, more effort on development the convex optimization methods has been put. Hence Nesterov and Nemirovskii proposed the Interior Point Method (IPM) algorithm (see e.g. [100]).

The main advantage of IPM algorithms is that they are the polynomial-time in the number of optimization variables - that fact comes from its construction. Also, the practical efficiency has been proved - the maximum number of iterations on usual does not exceed 50 and acts like it is not dependent of the input data.

Interior-point algorithm

Due to their mentioned advantages (the theoretical and computational efficiency - [100, 101]), IPM algorithms has been found very attractive for solving LMIs/SDPs. Below, the general sketch of IPM algorithm is presented ([102]).

Step 1 Construct a barrier function $\phi(x)$ that is well defined for strict feasible x and becomes $-\varepsilon$ (where $-\infty < -\varepsilon \ll 0$) only at the optimal value $x = x^*$.

Step 2 Generate a sequence $\{x^{(k)}\}$ so that

$$\lim_{k\to\infty} \phi(x^{(k)}) = -\varepsilon.$$

Step 3 Stop if $\phi(x^{(k)})$ is negative enough.

There arises the question of choosing the appropriate barrier function. It has to be convex inside the feasibility region of the considered problem and infinite outside it. The simplest choice is the following [102]

$$\phi(x) = -\log\det\big(F(x)\big) = \log\det\big(F^{-1}(x)\big).$$

The chosen barrier function is incorporated into the original objective function $f(x) = c^T x$ ($c \in \mathrm{R}^M$ is a given cost vector). However such a substitution produces the nonlinear objective function, but on the other hand the optimization problem becomes the unconstrained now and the problem appears to be easier to handle and finally to solve. The new objective function, which is to be minimized becomes then

$$\widehat{f}(x) = f(x) + u\phi(x) = c^T x - u\log\det\big(F(x)\big), \tag{3.12}$$

where the parameter $u > 0$ is to be selected. Now, any method for the nonlinear optimization can be applied. It relies on iterative producing the sequence of the solution still-better approximations ([85]) in the following manner

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}, \tag{3.13}$$

where $\Delta x^{(k)}$ defines the search direction and $t^{(k)} \geq 0$ is the size (length) of that step. The most common approach to solve such a problem is the Newton-based approach and in such a case $t^{(k)}$ denotes the gradient of $\widehat{f}(x)$, and $\Delta x^{(k)}$ denotes the inverse of the Hessian of $\widehat{f}(x)$.

There are many different implementations of the above algorithm. They differ in the ways how the gradient and Hessian are computed, the stop conditions and so on. As an example of the IPM algorithm the projective method [103, 104] can be considered. It has been implemented in Matlab LMI Control Toolbox and its brief description can be found in Section 3.9.1. This method uses the primal representation of the considered problem. However, there can be found a number of IPM algorithms employing primal-dual approach, see e.g. [102, 105, 89, 106].

Primal-dual IPM algorithm

Below the IPM algorithm, which uses the primal-dual approach to solve the SDP' given as (3.11), is presented (see [98, 97] and references therein).

Step 0 Set up the parameters $\gamma_1,\ \gamma_2\ \in\ (0,1)$, choose a starting point $(\widehat{\mathbf{Y}},\ \widehat{\mathbf{X}},\ \widehat{\mathbf{x}})$, where $\widehat{\mathbf{Y}},\ \widehat{\mathbf{X}},\ \widehat{\mathbf{x}}$ are defined as in (3.11) and set compute μ using the following equation

$$\mu = \frac{\widehat{\mathbf{Y}} \bullet \widehat{\mathbf{X}}}{n}. \tag{3.14}$$

Step 1 Compute the search direction $(d\widehat{\mathbf{Y}},\ d\widehat{\mathbf{X}},\ d\widehat{\mathbf{x}})$.

Step 1a Set $\mu := \gamma_1\mu$ and compute the residuals $r \in \mathrm{R}^{\hat{M}}$, $R \in \mathrm{R}^{\hat{M}\times\hat{M}}$

$$\begin{aligned} r_i &= \widehat{c}_i - \widehat{G}_i \bullet \widehat{\mathbf{Y}}, \quad (1 \le i \le \hat{M}), \\ R &= \widehat{G}_0 - \sum_{i=1}^{\hat{M}} \widehat{G}_i \widehat{x}_i - \widehat{\mathbf{X}}, \\ C &= \mu I - \widehat{\mathbf{Y}}\widehat{\mathbf{X}}. \end{aligned}$$

Step 1b Compute matrix $B \in \mathrm{R}^{\hat{M}\times\hat{M}}$ and the vector $s \in R^{\hat{M}}$

$$B_{ij} = \widehat{G}_i \bullet \widehat{\mathbf{Y}}\widehat{G}_j\widehat{\mathbf{X}}^{-1}, \quad (1 \le i \le \hat{M}), (1 \le j \le \hat{M}), \tag{3.15}$$

$$s_i = r_i - \widehat{G}_i \bullet (\widehat{\mathbf{Y}}C - R)\widehat{\mathbf{X}}^{-1}, \quad (1 \le i \le \hat{M}). \tag{3.16}$$

In general, B is fully dense positive definite symmetric matrix [107].

Step 1c Solve the Schur complement equation system $Bd\widehat{\mathbf{x}} = s$ to find $d\widehat{\mathbf{x}}$.

Step 1d Compute $d\widehat{\mathbf{X}}$ and $d\widehat{\mathbf{Y}}$ from $d\widehat{\mathbf{x}}$

$$\begin{aligned} d\widehat{\mathbf{X}} &= R - \sum_{i=1}^{\hat{M}} \widehat{G}_i d\widehat{x}_i, \\ d\widetilde{\widehat{\mathbf{Y}}} &= (C - \widehat{\mathbf{Y}}d\widehat{\mathbf{Y}})\widehat{\mathbf{X}}^{-1}, \\ d\widehat{\mathbf{Y}} &= (d\widetilde{\widehat{\mathbf{Y}}} + d\widetilde{\widehat{\mathbf{Y}}}^T)/2. \end{aligned}$$

Step 2 Determine the largest step size $\alpha_P,\ \alpha_D$ in the search direction $(d\widehat{\mathbf{Y}},\ d\widehat{\mathbf{X}},\ d\widehat{\mathbf{x}})$

$$\begin{aligned} \alpha_P &= -\lambda_{min}^{-1}(\sqrt{\widehat{\mathbf{Y}}}^{-1} d\widehat{\mathbf{Y}} \sqrt{\widehat{\mathbf{Y}}}^{-T}), \\ \alpha_D &= -\lambda_{min}^{-1}(\sqrt{\widehat{\mathbf{X}}}^{-1} d\widehat{\mathbf{X}} \sqrt{\widehat{\mathbf{X}}}^{-T}), \end{aligned}$$

where $\sqrt{H}$ denotes the matrix which satisfies $\sqrt{H}\sqrt{H}^T = H,\ H > 0$ and $\lambda_{min}(H)$ is the minimum eigenvalue of matrix $H > 0 \in R^{\hat{M}\times\hat{M}}$.

Step 3 Update the iteration point $(\widehat{\mathbf{Y}},\ \widehat{\mathbf{X}},\ \widehat{\mathbf{x}})$ using the search direction and the step size, and return to *Step 1*

$$\begin{aligned} \widehat{\mathbf{Y}} &= \widehat{\mathbf{Y}} + \gamma_2\alpha_P d\widehat{\mathbf{Y}}, \\ \widehat{\mathbf{X}} &= \widehat{\mathbf{X}} + \gamma_2\alpha_D d\widehat{\mathbf{X}}, \\ \widehat{\mathbf{x}} &= \widehat{\mathbf{x}} + \gamma_2\alpha_D d\widehat{\mathbf{x}}. \end{aligned}$$

The proper stop condition for the primal-dual algorithm presented above is to drive μ and the feasibility errors (primal, dual), approach close enough to 0 [97].

It is to note here that presented algorithm does not require the initial point ($\widehat{\mathbf{Y}}_0$, $\widehat{\mathbf{X}}_0$, $\widehat{\mathbf{x}}_0$) to be feasible (lay inside the feasibility region). This property appears to be extremely important when solving LMI where the main goal is to find the feasible solution starting from the unfeasible one.

3.8 Parallelization of the SDP algorithm

Note that in the presented algorithm some steps are especially time consuming, i.e.

- computing matrix B (*Step 1b*),
- performing the Cholesky factorization of B when solving the Schur complement equation system $B = d\widehat{\mathbf{x}}$ (*Step 1c*),
- computing matrices $d\widehat{\mathbf{Y}}$, $d\widehat{\mathbf{X}}$ (*Step 1d*),
- other computations involving dense matrices $\widehat{\mathbf{Y}}$, $\widehat{\mathbf{X}}$ (multiplication, addition).

Those matrix operations can be performed more efficiently when applying the parallel computations. Particularly, computations performed on B are paralleled. Since each entry of B is computed using (3.15), to compute the elements of i-th row of B the same matrix $\widehat{\mathbf{Y}}\widehat{G}_j\widehat{\mathbf{X}}^{-1}$ is used. Hence it is straightforward to see that every single row of B can be computed on a different CPU. After computing B, it is necessary to run the Cholesky factorization. This part is also performed using several CPUs. The parallel Cholesky factorization of B can be governed by ScaLapack.

There also arises the possibility to employ the parallel computing techniques in the computation of the search direction. In [98, 108], the proposition how to compute columns of $d\widehat{\mathbf{Y}}$ using the separate CPUs can be found.

3.9 LMI/SDP solvers

For last years many software packages enabling to solve LMI problems have been published. The increasing number of them is due to the rapid development of the theory (the further mutations of the basic IPM algorithm) on one hand and on the other the rapidly increasing the computational power of the current computers. The software packages, called the LMI/SDP solvers, available to solve those problems were implemented either as toolboxes and are included into the computation environments such as Matlab ([109]), Maple or Scilab ([110]) or as a independent libraries/binaries. The most popular package (solver) for the control problems is the Matlab LMI Control Toolbox ([44]), however there are plenty of others, which are also in use. Those include: Sdpa [89], Sdpt3 [111], Pennon [112], Maxdet [113], Sdpsol [114], Dspd [115], Moses [116], Maple package for Semidefinite Programming [117] and SeDuMi [118].

Some packages from the above list are available in versions, which can be used under several platforms. The best example here can be Sdpa, which is a independently installed application to work under Linux or MS Windows. Nevertheless, there is published the Matlab version of SDPA - SDPA-M [119]. The library for C/C++ programmers - SDPA-C [120] is available as well. There is also available the parallel computation implementation of SDPA - again in two versions as a independent LINUX binary - SDPARA [97, 98] or as a C library SDPARA-C [98]. The other example can be SCILAB LMI OPTIMIZATION PACKAGE [121], which is also available as a MATLAB toolbox.

The listed above software packages have been implemented to solve the specified problems (defined in the proper SDP form), which can differ e.g. MAXDET - determinant maximization problems, DSDP - combinatorics and PENNON - problems of convex and nonconvex nonlinear programming (aimed at large-scale problems with sparse data structure). Moreover, due to fact that they use different implementations of IPM algorithms, its efficiency can differ when particular problems are to be solved. Then the set of benchmarks has been published [122, 123] to provide the possibility to compare the effectiveness of selected packages for the particular kinds of problems.

Each of the described above packages requires the input data given in a specific form. Hence to make it easier to define the problems and to unify the way of doing it, YALMIP [124] has been released to act like a parser for the above packages. It supports most of the existing LMI solvers callable in MATLAB. YALMIP is a MATLAB application and its function is to allow to formulate the problem in the user-friendly way; next, to present it in the chosen SDP/LMI solver structure and finally, to call that solver. Thanks to its features, YALMIP is popular with the users, who can now easily choose the required solver and easily formulate the problem to be solved.

However there are plenty of software packages available, the following three are chosen to deal with the LMI problems in this dissertation:

- MATLAB LMI CONTROL TOOLBOX,
- SCILAB LMI OPTIMIZATION PACKAGE,
- SDPARA (a parallel computing version of SDPA).

3.9.1 Matlab LMI Control Toolbox

The MATLAB LMI CONTROL TOOLBOX was released in 1994. It implements the very efficient numerically Projective Algorithm of Nesterov and Nemirovski [100, 103, 104] which has the polynomial-time complexity. In this particular case, it means that the number of flops $Fl(\epsilon)$ required to compute an ϵ-accurate solution is upper bounded by

$$Fl(\epsilon) = \tau M^3 \log(\zeta/\epsilon),$$

where τ denotes the total row size of the LMI system (every LMI constraints defined), M denotes the total number of scalar decision variables, and ζ is a scaling factor. It is also important here that this implementation does not require an initial feasible point. This feature has to be treated as extremely valuable, when considering the kind of the problems investigated in the sequel of this dissertation.

There are defined three main kinds of problems that can be solved with MATLAB LMI CONTROL TOOLBOX (see [44]):

- feasibility problem (*feasp*),

- minimization of the linear function subject to LMI constraints (*mincx*),
- generalized eigenvalue minimization (*gevp*).

To solve each, the separate function has to be called.

The detailed description of the manner how to use the LMI Control Toolbox and programme the problems listed above can be found in Appendix A.1.

3.9.2 Scilab LMI Optimization Package

Scilab LMI Optimization Package ([121]) can be treated as a serious alternative to Matlab LMI Control Toolbox. Scilab is a computational environment, which uses similar to Matlab programming language and in some cases *m*-scripts and *m*-functions can be applied directly in Scilab. In comparison to Matlab, Scilab is competitive due to the following reasons:

- free of charge,
- provides the wide area of toolboxes (including the LMI solver),
- is still under development,
- is available for the whole set of Operational Systems (MS Windows, Linux, Unix) and hardware platforms (PC, Sun).

On the contrary to Matlab LMI Control Toolbox, Scilab LMI Optimization Package uses the primal-dual version of the IPM algorithm, which was originally presented in [125]. Its advantage, over Matlab LMI Control Toolbox, is that it allows explicitly to define the equality constraints (elimination of those is performed as the preprocessing step before the start pf the optimization procedure).

For the description how to use the Scilab LMI Optimization Package refer to Appendix A.2.

3.9.3 Parallel computing for the analysis and the synthesis of LRPs

Last years provided the increased computational needs. Hence the application of computers empowered to solve the large problems appeared to be quite natural. Such machines, able to solve large computational tasks (that the "average" PCs were not able to) at the time, were called supercomputers. Today the term "supercomputer" treats more the ability to compute than the construction of the machine. The first supercomputer of historical significance was Cray-1. Then it followed that it was reasonable to increase the number of CPUs, its speed, increase the size of operating memory and so on. Hence the parallel machines appeared on the market and today's supercomputers are parallel.

On the other hand, the rapid development of electronic technology makes the yesterday's supercomputer equal in abilities to today's desktop or even laptop computers. It is the straightforward conclusion of the Moore's law (see e.g. [126]) and puts into the question the reasonableness of spending the huge money for the single supercomputer, which will be obsolete in short period of time. Hence the idea of using supercomputers in form of one compact machine appeared to be too stiff. The Beowulf project (see [127]) proved that instead the buying the high-expensive compact super machines, it is more efficient to built a PC cluster, running the appropriate operating system and procedures involving the communication between the node PCs.

The natural choice of the operating system for PC clusters has been LINUX, which had several advantages over other available operating systems. LINUX has been being developed, hence it had been found very flexible and could be adopted easily to the required configuration. The other fact is that it has been free of charge and open source, hence everyone could trim it to the personal requirements. That includes also the PC cluster administrators.

To summarize, the PC clusters have the following advantages over alternatives:

- cost effective, thanks to using the market available components (PCs and network communication devices and free operating systems),
- robust to failures - every single hardware component can be replaced with the cheap replacement,
- easy to upgrade (by simply adding next computational nodes),
- high computational power.

Due to the above advantages the PC clusters become the popular powerful supercomputers.

As an interesting fact, it can be noted that the special set of benchmarks for testing the clusters was developed [128]. There is also a ranking of the most powerful at the time clusters (supercomputers) available (Top500 list at `http://www.top500.org`).

Generally, due to the components of cluster, there are two types defined, i.e. the homogenous, in which all PC nodes are the same; and heterogenous, where all or some, nodes can be different one from each other. The fact that in general clusters can be heterogenous can imply the situation when the clusters can be builded of plenty out of date and unused at the moment PCs.

SDPA (SDPARA)

The whole family of SDPA packages has been found to be an efficient software based on primal-dual IPM algorithm to solve SDPs. In general, SDPA (and what follows SDPARA) has been designed to solve large scale SDP, which can be sparse. Thanks to the parallel computing in SDPARA, relatively large SDPs/LMIs can be solved.

SDPA uses the IPM algorithm described in Section 3.7 and the parallel version of that algorithm with changes described in Section 3.8 has been used in SDPARA. It can be found (e.g. in [98, 97]) that parallelization of the computations concerns and finally deals with the bottleneck in the algorithm, i.e. computation of the search direction during the single iteration of the algorithm.

However, it does not suit exactly the terms of LMI control problems considered in this dissertation, but the possible income due to the application of the parallel computation methods is a important reason for trying to involve SDPARA to solve LMIs. Hence even if originally SDPARA has not been intended to solve LMIs, in Section 3.5 the method how to reformulate LMIs into the valid form of SDP has been presented. In the literature (see [97]), a survey of selected control problems and its solutions given in terms of SDPARA can be found.

On the contrary to two previously described software packages, SDPA (SDPARA) requires the input data given in the canonical form, instead of the matrix variables form. Another words, when for instance considering the 1D system stability investigation, instead of defining the matrix $P > 0$ and providing the system matrix A, here it is necessary to define a set of optimization variables $\widehat{x}_i$, vertex matrices $\widehat{G}_i$ (for notation refer to Section 3.5). The way of reformulating the original problem from LMI into SDP has been described in Example 3.3. Hence it is a straightforward conclusion that the size of the file containing the problem increases significantly because now instead of storing the dense matrices only (matrix A in Example 3.3) it has to contain all vertex matrices (G_i in Example 3.3). This can be treated as a serious limitation in application of the SDP solvers to solve the high dimensioned LMIs.

The other thing is a try to exploit the sparsity properties of LMIs. It is not so significant since considered LMIs on average appear to be dense but there arises one fact which can be used. Namely, when solving LMI control problems, there is always an assumption on positive definiteness of the matrix variable, i.e. $F(P) < 0,\ P > 0$, which is indeed treated as the following LMI

$$\begin{bmatrix} -F(P) & 0 \\ 0 & P \end{bmatrix} > 0.$$

Hence the above LMI can be treated as dense - 1st case or sparse - 2nd case, with two blocks ($F(P)$ and P). It is not huge improvement, but when considering the LMI problems of thousands decision variables, it is significant.

For the description how to use the SDPARA refer to Appendix B.3, where the 2D controller design problem MATLAB function to parse the LMI into the form of SDP problem suitable to solve by SDPARA has been attached.

The examples presented in this dissertation computed using SDPARA have been performed using the following nonhomogeneous cluster of 16 PCs, containing

- 8 PCs with CPU Intel Pentium IV HT, 3.00 GHz, 2 GB RAM (6001 bogomips),
- 8 PCs with CPU Intel Pentium IV 3.06 GHz, 1GB RAM (6006 bogomips).

The member PCs are the student laboratory computers available in laboratory rooms 405 and 406 at Institute of Control and Computation Engineering, University of Zielona Góra. The communication between the cluster nodes is governed by TCP/IP networking protocol. The member PCs of the cluster run LINUX SLACKWARE operating systems. The parallel computation environment is MPI (among plenty see e.g. [129, 130, 126]) with numerical libraries: ATLAS, CLAPACK, SCALAPACK, BLACS, SDPA and the SDP solver involving the parallel computing – SDPARA (see [97]). It will be abbreviated as CLUSTER in the sequel.

If it is not mentioned explicitly in the sequel, all computations run on CLUSTER have been performed using 16 member PCs.

Chapter 4

Analysis and synthesis of Linear Repetitive Processes

During last years properties of LRPs have been studied extensively. The most basic properties studied are stability concepts of this class of dynamical systems. Due to the special construction of LRP, two types of stability have been defined: asymptotic stability and stability along the pass.

This chapter refers the stability topics from [17, 18] but there is a serious extension of those due to applications of LMI in solving problems of analysis and synthesis of LRPs. In the sequel several LMI conditions for checking the asymptotic stability and stability along the pass of considered LRPs are presented and, what is more important, the list of stabilization conditions is given. These include: the "basic" controller design conditions for 1D equivalent model of LRP, the 1D model matching, decoupling of the dynamics and successive stabilization - for asymptotic stability. Respectively, for stability along the pass, the list of results presented here includes: "basic" controller design conditions, stabilization to prescribed stability margins, 2D model matching, output controller design. As aforementioned, those problems can be large dimensioned and require significant computational power to solve them. Hence some of them have been solved with methods involving the paralleling computing and the results are presented here as well.

It is to note that the results regarding the basics of the application of the LMI methods to analysis and synthesis of LRPs presented at the beginning of this chapter are provided [47, 46, 131, 49, 20]. The rest of results presented regard the extensions of those basic approaches, where the selected goals beyond the stability are requested to be addressed. Those are the original author's results and they have been published (or will be soon - see the references) in [25, 48, 60, 132] – Sections 4.1 – 4.4; [25, 133] – Section 4.10; [54, 70, 134] – Sections 4.7 and 4.8 and [135, 136, 137, 138] – Section 4.9.

4.1 Asymptotic stability – 1D system point of view

The concept, definition and conditions for asymptotic stability have been presented in Section 2.4.1. The asymptotic stability condition for (2.9)-(2.10) or (2.14)-(2.15) in terms of LMI is provided by the following theorem

Theorem 4.1 *[25] The LRP is asymptotically stable if and only if there exists $P > 0$ of the* *appropriate dimensions such that the following LMI holds*

$$D_0^T P D_0 - P < 0. \tag{4.1}$$

It is straightforward to see that (4.1) is equivalent to (2.29).

The above condition is applicable only for LRPs of (2.9)-(2.10) and (2.14)-(2.15) with boundary conditions defined in its simplest form (2.11) – discrete case and (2.16) – differential case. It is not applicable for the processes with more sophisticated boundary conditions and, what is more important here, it cannot be applied for generalized LRPs of (2.19)-(2.20). As it was mentioned in Section 2.4, for that class of LRPs the more accurate is the application of the 1D equivalent model of (2.24)-(2.25) and the conditions given for it. Hence the following LMI condition addresses this topic.

Theorem 4.2 *[25] The discrete LRP given in the 1D equivalent model of (2.24)-(2.25) asymptotically stable if and only if there exists $P > 0$ of the appropriate dimensions such that the following LMI holds*

$$\Phi^T P \Phi - P < 0. \tag{4.2}$$

Note that the proof for the above theorem and hence the equivalence between Theorems 2.2 and 4.2 can be presented exactly in the manner as for the Theorems 2.1 and 4.1.

Comparing the results of Theorems 4.1 and 4.2 to previously presented Theorems 2.1 and 2.2, it is straightforward to notice that they concern the same property. However, conditions given above (LMI conditions) posses the natural and easy ability to extend them for the synthesis problem (controller design). Nevertheless, regarding Theorem 4.2, there still continues one serious limitation. Remain Example 2.1. Due to the fact that 1D equivalent model, in the most considered cases, can be of the huge dimensions (high order model), the application of LMI would require involving the huge number of decision variables. For instance, for the condition of Theorem 4.2 it would be $M = (m\alpha)(m\alpha+1)/2$ and for data given in Example 2.1 the number of decision variables equals 500500 which is huge and in practice out of range for today's computers (even when considering the PC clusters). Nevertheless, there exist sufficient numerically software packages (LMI/SDP solvers) which can be used to solve relatively large problems formulated in the form of LMI of (4.2) (refer to Section 3.5 from the previous chapter). They can be used in the cases, when single computers are too slow or have too less operational memory to solve the problem. Hence when the problem appears to be too large to be solved it using the single computer, it is possible to try to solve it using the PC cluster with parallel computing techniques, which computational abilities can be increased by simply connecting more nodes to the cluster.

4.2 The controller design towards asymptotic stability

In terms of the controller design, a key task is to ensure asymptotic stability under the control action. It is worth to mention now that in the discrete case of (2.9)-(2.10) with the boundary conditions of (2.11), the asymptotic stability alone is rarely required. It is due to the fact that for this model, it is relatively easy to define and implement the "stronger" demand, i.e. the stability along the pass (this concept will be considered in the sequel of this chapter). On the other hand, in general, for any other member of the class of LRPs considered in this dissertation, for which it is very hard to define the conditions for stability along the pass to hold, it is natural to settle for asymptotic stability only. What is more, according to the fact that the structures of those models are not so clear as for (2.9)-(2.10) (e.g. there is no equivalent condition to (2.29) for LRP of (2.19)-(2.20)), it is straightforward to apply 1D equivalent model of LRP and stability theory regarding this model. In the context of controller design towards asymptotic stability, Theorem 4.2 is accurate to provide the required solutions.

Hence for any discrete LRP, where the 1D equivalent model is used, the task is now defined as constructing the controller to apply in the control law of the form

$$U(l) = KV(l), \tag{4.3}$$

such that $r(\Phi + \Delta K) < 1$. Note again that in implementation terms the design exercise requires computations with potentially very large dimensioned matrices.

The direct application of the synthesis method presented in [81] for the classical discrete 1D systems, let us to present the result of Theorem 4.2 in its closed loop version as follows

Theorem 4.3 *[25] Suppose that the discrete LRP is considered in terms of its 1D equivalent model of (2.24)-(2.25) with the appropriate matrices and the control law of the form (4.3) is used. Then the closed loop asymptotic stability holds if and only if there exist matrices $P > 0$ and N of the appropriate dimensions such that the following LMI holds*

$$\begin{bmatrix} -P & P\Phi^T + N^T\Delta^T \\ \Phi P + \Delta N & -P \end{bmatrix} < 0. \tag{4.4}$$

Also if this condition holds then K in (4.3) can be computed as

$$K = NP^{-1}. \tag{4.5}$$

The above result is adopted from the "classical" 1D systems theory and its detailed description can be found in [81].

A potential difficulty in applying Theorem 4.3 may arise since P is a LMI decision matrix and simultaneously is used to compute K. The next result seems to be better in this respect and is based on the 1D case as in [80], but on the other hand, it increases the total number of decision variables to be found.

Theorem 4.4 *[25] Suppose that the discrete LRP is considered in terms of its 1D equivalent model of (2.24)-(2.25) with the appropriate matrices and the control law of the form (4.3) is used. Then the closed loop asymptotic stability holds if and only if there exist matrices $P > 0$, G and N of the appropriate dimensions such that the following LMI holds*

$$\begin{bmatrix} -P & \Phi G + \Delta N \\ G^T\Phi^T + N^T\Delta^T & P - G - G^T \end{bmatrix} < 0. \tag{4.6}$$

Also if this condition holds, a stabilizing K is given by

$$K = NG^{-1}. \tag{4.7}$$

According to the possible large dimensions of considered problems there arise two main problems in solving the controller design problem, i.e.

- handling with the matrices of large dimensions (see Example 2.1) and
- rapid increasing with the number of the decision variables (and hence the time) necessary for the optimization procedure to converge to the solution.

One possible remedy for those obstacles is to apply the "strong enough" computational machines with the appropriate software installed. However, it should be pointed out that for the problems considered here single computers seem to be too weak to deal with. That's why, it is suggested to use the PC clusters and take the advantage of the parallel computing.

The other way to handle with those problems is to try to formulate the task in the form which can simplify it (e.g. introduce the additional constraints on the decision matrices, which would decrease the total number of the decision variables or/and to apply a specially developed algorithms dealing with the concrete situation). However such a simplification of the problem causes that the originally necessary and sufficient condition becomes only the sufficient one.

To highlight the problems presented above consider the following example. Here, it is assumed that the structure of the controller is restricted to become the block diagonal.

Example 4.1 *Consider the model of the generalized LRP of (2.19)-(2.20) with* $\alpha = 5$

$$A=\begin{bmatrix} -0.63 & 0 & 0 & -0.18 \\ -0.1 & -0.28 & 0 & 0.7 \\ 0.87 & 0.11 & -0.05 & -0.02 \\ 0.08 & 0 & 0 & -0.17 \end{bmatrix},\ B=\begin{bmatrix} 0.29 & 0 \\ 0 & 0.74 \\ 0 & -0.74 \\ 0 & -0.27 \end{bmatrix},\ C=\begin{bmatrix} 0 & 0.42 & 0 & 0 \\ 0.9 & -0.1 & -0.21 & 0.15 \\ -0.22 & 0.04 & 0 & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} 0.34 & 0.19 \\ -0.4 & 0.02 \\ -0.42 & 0.25 \end{bmatrix},\quad \mathbf{B}=\left[\begin{array}{c|c|c|c} B_0 & B_1 & \dots & B_4 \end{array}\right]=\left[\begin{array}{ccc|ccc|ccc|ccc|ccc} 0.95 & 0.69 & 0.54 & 0.21 & -0.11 & 0 & -0.62 & 0.62 & 0.3 & -0.82 & 0.76 & 0.71 & 0.75 & -0.95 & 0 \\ -0.33 & 0.36 & 0 & 0.91 & 0.67 & 0 & 0 & 0.86 & 0.21 & 0 & 0 & 0 & 0.86 & 0.95 & -0.08 \\ -0.06 & -0.95 & 0 & -0.82 & -0.47 & 0 & -0.88 & 0.4 & -0.55 & 0.55 & 0 & -0.53 & 0.2 & 0.13 & 0.41 \\ 0.67 & 0.34 & 0 & 0.79 & 0.07 & -0.76 & 0 & 0 & 0.61 & -0.51 & 0.56 & 0 & 0.61 & 0.06 & 0 \end{array}\right],$$

$$\mathbf{D}=\left[\begin{array}{c|c|c|c} D_0 & D_1 & \dots & D_4 \end{array}\right]=\left[\begin{array}{ccc|ccc|ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 1.49 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.9 & 0.67 \\ -0.14 & 0.9 & 1.15 & 1.26 & 0 & 1.69 & 0.35 & 0.17 & -0.35 & -0.03 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.46 & -1.71 & 0.12 & -0.47 & 0 & -1.04 & 0 & -0.64 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

It is assumed that controller takes the following structure

$$K = \begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 \\ K_{21} & K_{22} & K_{23} & 0 & 0 \\ 0 & K_{32} & K_{33} & K_{34} & 0 \\ 0 & 0 & K_{43} & K_{44} & K_{45} \\ 0 & 0 & 0 & K_{54} & K_{55} \end{bmatrix}.$$

Such a choice is dedicated by the necessity to restrict the number of the pass profile vectors which are used to stabilization. In this case, at every point on the pass, three pass profile vectors from the previous pass are used (with the exception of the first and the last point on the pass). Such a controller can be obtained via choosing the matrices G and N (P is assumed to be full symmetric matrix) in the LMI of (4.6) as follows

$$N = \begin{bmatrix} N_{11} & N_{12} & 0 & 0 & 0 \\ N_{21} & N_{22} & N_{23} & 0 & 0 \\ 0 & N_{32} & N_{33} & N_{34} & 0 \\ 0 & 0 & N_{43} & N_{44} & N_{45} \\ 0 & 0 & 0 & N_{54} & N_{55} \end{bmatrix}, \; G = \begin{bmatrix} G_1 & 0 & 0 & 0 & 0 \\ 0 & G_2 & 0 & 0 & 0 \\ 0 & 0 & G_3 & 0 & 0 \\ 0 & 0 & 0 & G_4 & 0 \\ 0 & 0 & 0 & 0 & G_5 \end{bmatrix}.$$

The application of Theorem 4.4 provides the following controller matrix

$$K = \left[\begin{matrix} 0.2630 & -1.1896 & -1.5577 & 3.3029 & -0.8337 & 3.2820 & 0 \\ 0.9336 & -1.8058 & -2.7157 & 3.1879 & -0.5154 & 0.9308 & 0 \\ -1.1407 & -5.6974 & -5.1293 & -0.7307 & -0.7983 & -2.3297 & -0.0222 \\ 0.0117 & 2.7425 & 4.2268 & -3.2095 & 0.4962 & -4.3639 & 2.8251 \\ 0 & 0 & 0 & 7.3285 & -0.8166 & 7.9974 & -0.9771 \\ 0 & 0 & 0 & 3.9953 & -0.2126 & 2.8505 & 0.9518 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.9468 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2601 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \right.$$

$$\left. \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.3095 & 0.7040 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.7240 & 2.8575 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2603 & -1.9817 & 0.1295 & 0.0857 & 0.4195 & 0 & 0 & 0 \\ 0.0675 & -0.76 & 0.0689 & -0.1241 & 0.4053 & 0 & 0 & 0 \\ -0.1427 & -0.6172 & -0.3033 & 0.0863 & 0.2983 & -0.5609 & -4.6653 & -1.1218 \\ -1.7105 & 0.9546 & 1.2597 & -0.9765 & -0.7081 & -5.3617 & -32.0398 & -8.9437 \\ 0 & 0 & -1.0893 & 0.6052 & 0.8259 & -0.2293 & -5.2978 & -1.2641 \\ 0 & 0 & -1.1578 & 1.1583 & 1.0251 & 3.0619 & 13.5926 & 3.8862 \end{matrix} \right].$$

Remark 4.1 *Considering the result of this example, the following property should be pointed out. Namely, it is to note that the application of the LMI condition of Theorem 4.4 provides the feasible solution, when the application of the condition given in Theorem 4.3 appears to be unsolvable (without deciding about the feasibility or infeasibility of the problem). This phenomena can be explained by noting the structure of the considered LMIs. As it was mentioned, in (4.6) the Lyapunov matrix and the matrix which is used to compute the controller are different. Hence the conservativeness of this condition should be smaller (the problem should be solvable easier) than the other. The above example proves this presumption.*

Concluding, in problems stated as above when e.g. the controller has the structure set a'priori Theorem 4.4 proves its better applicability over Theorem 4.3. Nevertheless, from the theoretical point of view, both those conditions are equivalent (see the proof of Theorem 4.4 in [25]).

Remark 4.2 *Note that Example 4.1 can be treated as academic since the assumed pass length is small. Nevertheless, it is ideal example to show how many decision variables are required. In this particular case* $\Phi \in \mathrm{R}^{15\times 15}$, $\Delta \in \mathrm{R}^{15\times 10}$ *and the total number of variables which are involved in the LMI equals* 243*. Remain that here it is assumed that the controller is of the constrained structure (for the full structure controller the total number of decision variables equals* 495*). What is more, if the pass length would be assumed to be equal* 6 *(increased by* 1*) the total number of the decision variables would equal* 286 *for the constrained controller and* 584 *for the full controller. This illustrates that increasing one parameter by* 1*, causes increasing the total number of the variables more than* 10% *(and hence the size to be solved). It is straightforward to see that when considering the real lengths of passes (say* 50 *or more), the total number of variables to handle is huge, even when considered condition limits that number by putting the constraints on the controller structure. Hence to handle such huge problems, it is necessary to employ either the simplifications of the considered problem (e.g. as presented) or to apply the parallel computing.*

Example 4.2 *To check the effectiveness of the PC clusters used to solve the controller design problems, some tests have been done. Here, the generalized LRPs of (2.19)-(2.20) are considered. The 1D equivalent model is used. Hence in this case the system matrix* Φ *is of the size* $m\alpha \times m\alpha$ *and the input matrix* Δ *- of* $m\alpha \times r\alpha$*, respectively. Hence it is abbreviated that the size of the problem is* $m\alpha \times r\alpha$*. The LMI condition of Theorem 4.3 is considered, however it is also possible to perform such tests for LMI of Theorem 4.4.*

To perform the tests, a set of models of (2.19)-(2.20) has been chosen in such a manner that $A \in \mathrm{R}^{5\times 5}$, $B \in \mathrm{R}^{5\times 1}$, $C \in \mathrm{R}^{5\times 3}$, $D \in \mathrm{R}^{3\times 1}$ *are the same in all cases and the pass length (and hence sizes of* B_j *and* D_j*) is increased by* 2 *starting from* $\alpha = 4$*. Hence the resulting 1D equivalent models are of the sizes* 12×4, 18×6, $24 \times 8, \ldots, 120 \times 40$ *(using the aforementioned abbreviated notation). The synthesis tasks for those models have been solved to estimate the computational complexity of that task.*

Table 4.1 presents the sizes of problems and times required to solve those on CLUSTER. Figure 4.1 shows the computational complexity of the considered synthesis problem.

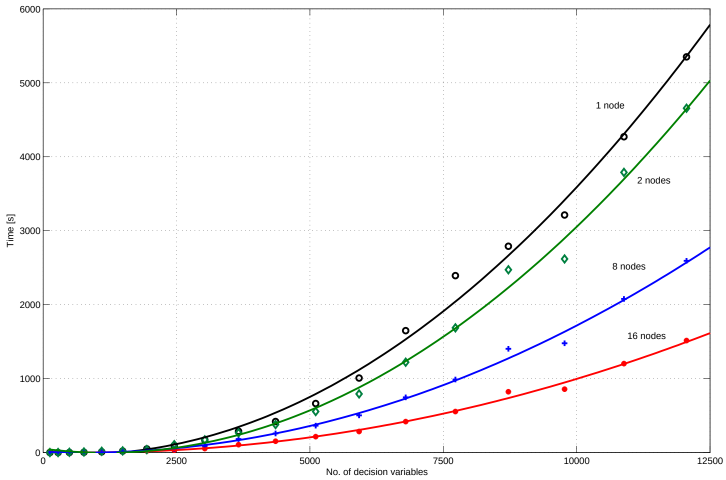


Figure 4.1. The computational complexity of the considered synthesis problems (approximated with the 2nd degree polynomial)

$m\alpha \times r\alpha$	No. of decision variables	1 node Time [s]	2 nodes Time [s]	4 nodes Time [s]	8 nodes Time [s]	12 nodes Time [s]	16 nodes Time [s]
12×4	126	0.1	0.3	0.4	1.2	1.6	2
18×6	279	0.4	1	1	1	2	2
24×8	492	2	3	4	3	4	4
30×10	765	4	8	7	6	6	6
36×12	1098	10	17	12	9	11	8
42×14	1491	21	23	23	17	16	14
48×16	1944	49	41	39	27	23	21
54×18	2457	96	107	99	50	40	32
60×20	3030	171	172	117	90	71	54
66×22	3663	289	268	234	182	144	107
72×24	4356	420	379	331	259	202	153
78×26	5109	663	555	472	364	284	215
84×28	5922	1008	793	658	503	395	285
90×30	6795	1649	1222	990	746	580	419
96×32	7728	2393	1686	1331	988	768	555
102×34	8721	2790	2472	1918	1404	1094	822
108×36	9774	3212	2618	2024	1478	1160	857
114×38	10887	4270	3788	2878	2078	1617	1203
120×40	12060	5350	4656	3636	2592	2025	1514

Table 4.1. The sizes of the considered problems and times required to solve those on cluster using different number of nodes in the cluster

The considered problems have been solved on cluster with 1, 2, 4, 6, 8, 10, 12, 14 *and finally* 16 *nodes. Points shown in Figure 4.1 shows the time versus number of variables dependence. Points denoted as* $\circ$ *present results obtained with only* 1 *node in the cluster,* $\diamond$ - 2 *nodes,* + - 8 *nodes and* $*$ - 16 *nodes, respectively. The solid lines present the approximated polynomials of the* 2*nd degree. Figure 4.2 shows the approximated polynomial of the* 1*st degree. It is to see that the linear function does not suit sufficiently well the data, which is to be approximated but* 2*nd order polynomial is sufficiently good and hence the computational complexity can be presented as the* 2*nd degree polynomial of the size (number of decision variables).*

It is to note that the increase of the number of nodes to the maximal number of 16 *in the cluster does not change (decrease) the degree of the polynomial approximating the complexity. However, it decreases significantly (almost* 4 *times) the leading coefficient in the polynomial, which influences the speed of the computations. The other aspect of using the clusters is the dependence which arises between the number of the nodes in the cluster and speed-up of the computations. Hence the following figures show the speed-up of the computations related to the number of the nodes in the cluster.*

It is to note that the increase of the nodes in the cluster provides the sufficient results (the acceleration of the computations) only for the large size problems. Figure 4.3 shows the acceleration for the synthesis problem of 12060 *decision variables (large size problem) and it is straightforward to see that using* 16 *nodes in the cluster provides* 4 *times faster computations.*

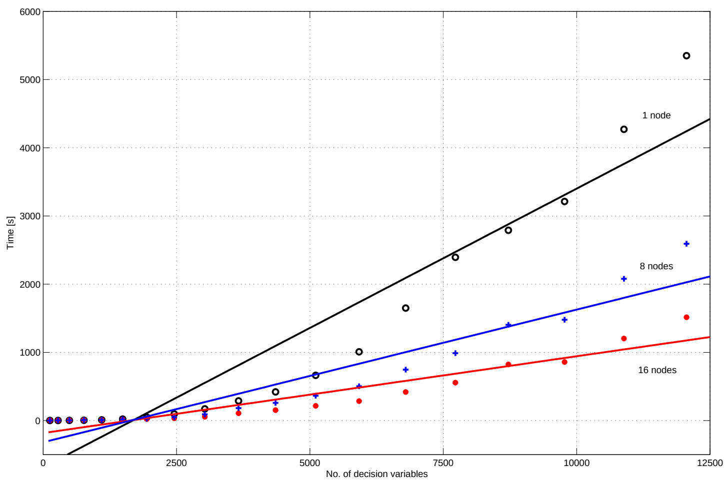


Figure 4.2. The computational complexity of the considered synthesis problems (approximated with the linear functions)

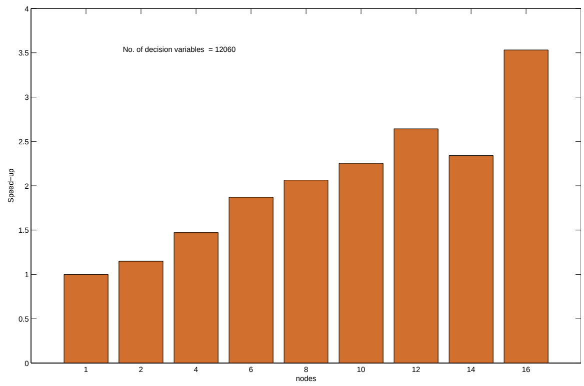


Figure 4.3. Speed-up for the large size problem (12060 variables)

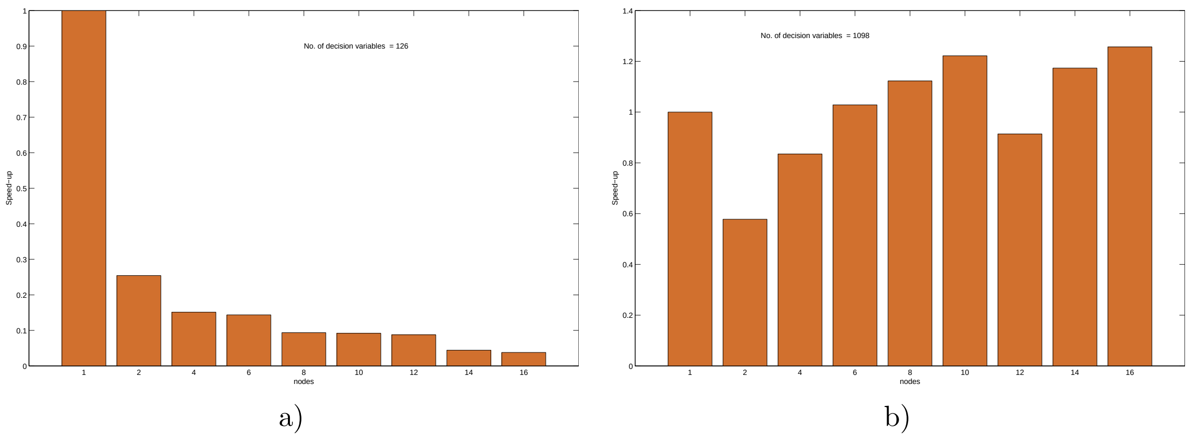


Figure 4.4. Speed-up for the small (126 variables) (a) and medium (1098 variables) (b) size problems

On the other hand, it can be concluded that for the small size problems it is not recommended to use a cluster. For the small problems, the cost of the synchronization between nodes gets away all the income from the parallel computing - see Figure 4.4 a). For the cluster tested in this particular case, the size of the problem, from which the application of the clusters appeared to be profitable equals (approximately) 1000. *It can be concluded that for the problems of the smaller sizes it is not justified to use clusters of* 16 *nodes and for the larger problems, it is justified (refer to Figure 4.4 b)).*

4.3 Stability along the pass – 2D systems point of view

As it has been described in Chapter 2, the significant difference between asymptotic stability and stability along the pass is that, when the first is one of the necessary conditions for the second one. There can be the LRP, in which asymptotic stability holds and simultaneously stability along the pass does not hold. In such a case, the limit profile, however exists, is unstable. The reason why asymptotic stability does not guarantee a limit profile, which is "stable along the pass" is due to the fact that it does not concern the dynamics along the pass (along p). To see that this property does not guarantee that the limit profile has "acceptable" along the pass dynamics refer to Example 2.2.

4.3.1 Discrete LRPs

Theorem 2.3 defines the necessary and sufficient conditions for stability along the pass, however it has to be outlined that the practical applicability of those conditions is really small. To provide the condition, which could be used in practice (and in the sequel synthesis), define the following matrices from the state-space model (2.9)-(2.10)

$$\widehat{A}_1 = \begin{bmatrix} A & B_0 \\ 0 & 0 \end{bmatrix}, \quad \widehat{A}_2 = \begin{bmatrix} 0 & 0 \\ C & D_0 \end{bmatrix}. \tag{4.8}$$

Then it is possible to present the following result.

Theorem 4.5 *[47] The discrete LRP described by (2.9)-(2.10) is stable along the pass if there exist matrices $P > 0$ and $Q > 0$ satisfying the following LMI*

$$\begin{bmatrix} \widehat{A}_1^T P \widehat{A}_1 + Q - P & \widehat{A}_1^T P \widehat{A}_2 \\ \widehat{A}_2^T P \widehat{A}_1 & \widehat{A}_2^T P \widehat{A}_2 - Q \end{bmatrix} < 0 \tag{4.9}$$

Remark 4.3 *Note that result of Theorem 4.5 provides only the sufficient condition for the stability along the pass. Recently, there was published a paper by Bliman [52] regarding the decreasing of the conservativeness of the given LMI condition and getting "closer" to the sufficient and necessary condition for stability of 2D systems in RM form. Due to the similarities of the RM and discrete LRP, mentioned formerly, it can be applied as well for LRPs. This new approach relies on sequentially increasing the state vector by the next delayed state vectors till the feasible solution of the appropriate LMI condition is found or the certificate of infeasibility is given. This approach is not considered here due to the fact, it has the limited applicability in terms of the controller design.*

The LMI condition for stability along the pass of discrete LRPs equivalent to (4.9) can be stated as follows

Theorem 4.6 *[20] The discrete LRP described by (2.9)-(2.10) is stable along the pass if there exist matrices $Y > 0$ and $Z > 0$ satisfying the following LMI*

$$\begin{bmatrix} Z-Y & 0 & Y\widehat{A}_1^T \\ 0 & -Z & Y\widehat{A}_2^T \\ \widehat{A}_1 Y & \widehat{A}_2 Y & -Y \end{bmatrix} < 0.$$

Another, known approach to stability along the pass investigation is based on using the block diagonal decision matrices of appropriate dimensions and the following theorem can be stated.

Theorem 4.7 *[20] The discrete LRP described by (2.9)-(2.10) is stable along the pass if there exists matrix $W = diag(W_1, W_2) > 0$ satisfying the following LMI*

$$\Upsilon^T W \Upsilon - W < 0,$$

where Υ is so-called plant matrix and has been defined in (2.18).

4.3.2 Differential LRPs

The similar analysis can be done for the case of differential LRP of (2.14)-(2.15). Here, only the most often applicable condition for the stability along the pass in term of LMI is given.

Theorem 4.8 *[49] The differential LRP is stable along the pass if there exist matrices $Y > 0$ and $Z > 0$ satisfying the following LMI*

$$\begin{bmatrix} YA^T + AY & B_0 Z & YC^T \\ ZB_0^T & -Z & ZD_0^T \\ CY & D_0 Z & -Z \end{bmatrix} < 0. \tag{4.10}$$

4.4 2D controller design towards stability along the pass

In the case, when considered LRP appears to be unstable along the pass, it is possible to try to stabilize it via the current state/output based feedback control loop. The goal now is to provide such a control sequence that ensures the closed loop system is stable along the pass.

4.4.1 Discrete LRPs

The synthesis is done similarly as it had a place for the asymptotic stability. Define a control law of the following form over $0 \leq p \leq \alpha - 1,\ k \geq 0$ ([47])

$$u_{k+1}(p) = K_1 x_{k+1}(p) + K_2 y_k(p) := K \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix}, \tag{4.11}$$

where K_1 and K_2 are appropriately dimensioned controller matrices to be designed. In effect, this control law uses feedback of the current trial state vector (which is assumed to be available for use) and 'feedforward' of the previous trial pass profile vector. Note that in repetitive processes

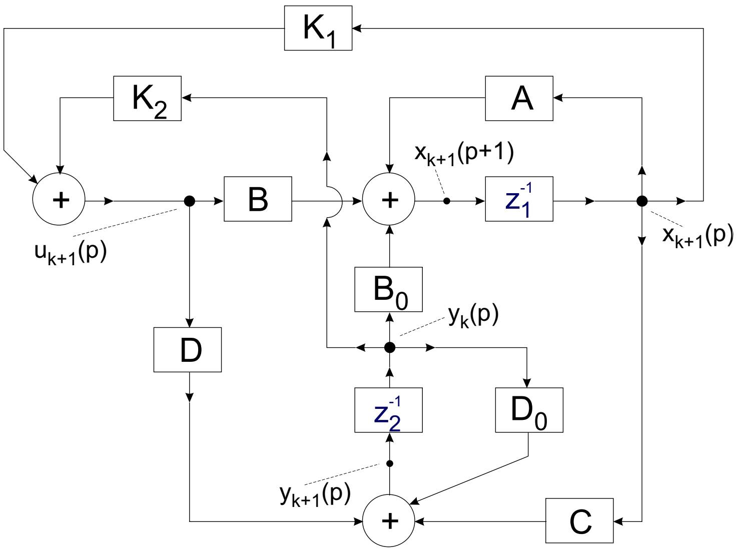


Figure 4.5. State/output control scheme

the term feedforward is used to describe the case where (state or pass profile) information from the previous pass (or passes) is used as (part of) the input to a control law applied on the current pass, i.e. to information which is propagated in the pass to pass (k) direction. The block scheme for the closed loop process under the control law of (4.11) is shown in Figure 4.5.

This control law has a physical meaning for practical applications of discrete LRPs and the following result uses the LMI setting to give a controller design algorithm, which can be easily implemented.

Theorem 4.9 *[47] Suppose that a discrete LRP of (2.9)-(2.10) is subject to a control law of the form (4.11). Then the closed loop system is stable along the pass if there exist matrices* $Y > 0$, $Z > 0$, *and* N *such that the following LMI holds.*

$$\begin{bmatrix} Z - Y & 0 & Y\widehat{A}_1^T + N^T\widehat{B}_1^T \\ 0 & -Z & Y\widehat{A}_2^T + N^T\widehat{B}_2^T \\ \widehat{A}_1 Y + \widehat{B}_1 N & \widehat{A}_2 Y + \widehat{B}_2 N & -Y \end{bmatrix} < 0, \tag{4.12}$$

where $\widehat{A}_1$, $\widehat{A}_2$ *are defined as in (4.8) and*

$$\widehat{B}_1 = \begin{bmatrix} B \\ 0 \end{bmatrix}, \; \widehat{B}_2 = \begin{bmatrix} 0 \\ D \end{bmatrix}. \tag{4.13}$$

If (4.12) holds, then a stabilizing K *in the control law (4.11) is given by*

$$K = NY^{-1}. \tag{4.14}$$

It is also possible to define the LMI controller design condition for the result of Theorem 4.7. Hence the closed loop version of Theorem 4.7 becomes the following one.

Theorem 4.10 *[47] The discrete LRP described by (2.9)-(2.10) is stable along the pass under the control law defined as (4.11) if there exist matrices* $W = diag(W_1, W_2) > 0$ *and* N *of* *appropriate dimensions satisfying the following LMI*

$$\begin{bmatrix} -P & W\Upsilon^T + N^T\widehat{\Omega}^T \\ W\Upsilon + \widehat{\Omega}N & -W \end{bmatrix} < 0, \tag{4.15}$$

where Υ *is so-called plant matrix and has been defined in (2.18) and*

$$\widehat{\Omega} = \begin{bmatrix} B & 0 \\ 0 & D \end{bmatrix}, \; N = \begin{bmatrix} N_1 & N_2 \\ N_1 & N_2 \end{bmatrix}.$$

When the above LMI holds, then the 2D controller matrix can be computed as

$$\widehat{K} = \begin{bmatrix} K_1 & K_2 \\ K_1 & K_2 \end{bmatrix} = NW^{-1}.$$

Naturally, there arises the question about the differences between conditions given in Theorem 4.9 and Theorem 4.10 (or 4.6 and 4.7). Note that those conditions are the natural consequence of the ability to present LRP in the form of FM or RM, i.e. those LMIs are the appropriately adopted stability conditions for RM or FM. From the theoretical point of view, there is no difference (see [47]) but when considering the synthesis problems of large dimensioned LRPs, it becomes clear that condition given in Theorem 4.10 (Theorem 4.7) can be solved faster than others. It comes from the fact that the time required to solve the LMI is polynomial function of the number of decision variables involved in computations. Hence the following relation is supposed straightforward to hold: the smaller number of decision variables, the less time required for obtaining the solution. Numbers of the decision variables required for the considered approaches are listed in Table 4.2.

Theorem 4.9	Theorem 4.10
$M = (m+n)(m+n+1) + r(m+n)$	$M = m(m+1)/2 + n(n+1)/2 + r(m+n)$
Theorem 4.6	Theorem 4.7
$M = (m+n)(m+n+1)$	$M = m(m+1)/2 + n(n+1)/2$

Table 4.2. Total numbers of decision variables used in considered Theorems

However, for the small size problems that difference is not significant, for the large size problems (say $M > 1000$) it becomes significant.

Concluding, it is easy to note that the same problem of analysis (synthesis) can be treated and solved differently. Hence the solution for that problem can be obtained sooner or later.

On the other hand, the condition given in Theorem 4.9 is searching the solution over the larger possible area (more decision variables) than Theorem 4.10 and hence that fact can cause that the solution of the problem is found, when the competing condition fails.

The following examples regard the difference between those conditions.

Example 4.3 *The aim here is to highlight how for conditions of Theorems 4.9 and 4.10 the total number of variables increases when the size of considered LRP is increased. There is also presented the total time required to finish the computations run using CLUSTER.*

It is to be seen that the application of the LMI condition given in Theorem 4.9 involves almost twice more decision variables than the LMI condition given in Theorem 4.10. It is

	Theorem 4.9		Theorem 4.10	
$n \times m \times r$	Total no. of decision vars	Total time [s]	Total no. of decision vars	Total time [s]
$10 \times 6 \times 8$	400	8.5 (8 - 8 PCs)	204	3.5 (1.5 - 8 PCs)
$12 \times 6 \times 9$	504	11 (11 - 8 PCs)	261	4 (2.5 - 8 PCs)
$15 \times 8 \times 10$	782	22	386	5 (4.99 - 8 PCs)
$20 \times 10 \times 15$	1380	52	715	12
$30 \times 10 \times 20$	2440	141	1320	38
$40 \times 15 \times 30$	4730	503	2590	116
$50 \times 20 \times 40$	7770	1363	4285	365

Table 4.3. Comparison of the results of Theorems 4.9 and 4.10

straightforward to see that the second condition is less memory demanding. Hence the same problem can be solved faster by simply applying the appropriate condition.

Note that for the small size problems, the total times required to finish the computations using both conditions are similar. With the growth of the size of problem, the total time values start to differ for the considered conditions.

Figure 4.6 shows the dependance how the total number of decision variables increases, when the size of problem grows for both considered conditions.

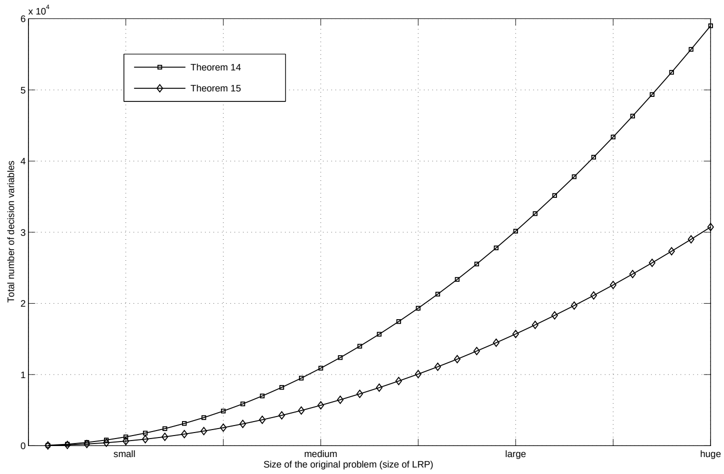


Figure 4.6. The growth of the total number of variables in Theorems 4.9 and 4.10

The above example suggests that the application of Theorem 4.10 provides better results than Theorem 4.9. However, it is to be noted that condition of Theorem 4.10 can cause the numerical problems and hence can fail. To see that, refer to Example 4.4. It can be explained by the fact that Theorem 4.9 searches the feasible solution over the larger space of possible solutions than

Theorem 4.10, hence when that second LMI fails, the first is still able to find the suitable solution. Nevertheless, the "price" which is to be paid for that has been presented in Example 4.3.

Example 4.4 *Consider the following model of the discrete LRP of (2.9)-(2.10)*

$$A = \begin{bmatrix} -1.2 & 0.7 & -1.1 & 1.3 & 1.7 & -0.4 & 0.6 & -1.4 & 1.9 & 0.9 \\ 1.7 & 1.8 & -0.1 & 1.8 & 0.8 & -0.7 & -1 & -1.2 & -1.5 & 0.2 \\ -2.1 & -0.7 & 0.1 & -1.3 & -0.3 & 1.9 & -1.5 & -1.2 & -1.8 & 1.9 \\ 0.4 & 0.3 & 1.2 & 0.7 & 0.8 & 1.4 & 0.6 & 0.6 & -1.6 & -0.7 \\ 0.2 & -1.6 & -1.3 & 1.8 & 0.5 & -1 & 1.9 & -1.9 & -1.4 & 0.9 \\ 0.6 & -1.4 & 1.7 & -0.7 & -0.8 & -1.9 & 1.3 & -1.1 & 1.7 & 1.9 \\ -0.8 & -0.9 & 1.8 & 0.4 & 1.5 & -2.1 & 1.8 & 0.7 & -1.5 & 0.3 \\ -1.1 & 0.2 & -2 & 0.5 & -1.6 & 0.3 & -0.8 & -0.8 & 0.5 & 1.6 \\ -0.4 & -0.1 & 1.1 & -2.1 & -0.9 & 1 & -1 & -0.8 & -1 & 1 \\ -0.8 & 1.9 & 1.9 & 2 & -1.7 & 1.3 & 0.2 & 0.9 & -1.2 & -0.5 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.6 & 0.7 & -0.3 & -0.5 & -0.1 & -0.5 & -0.3 & -0.3 \\ -0.7 & -1 & -0.9 & -0.3 & 0.7 & -0.3 & -0.2 & 0 \\ 0.3 & 0.5 & 0.4 & 0.5 & 0.7 & 0.8 & 0.8 & 0.9 \\ -1 & 0.7 & 0.9 & 0.3 & -0.1 & -0.4 & 0.5 & -0.3 \\ -0.3 & 0.5 & -0.7 & 0.9 & 0.6 & -0.5 & 0.8 & -0.5 \\ -0.4 & 0.9 & -0.2 & 0.7 & 0.3 & -0.1 & -0.1 & 0.2 \\ -0.9 & 0.3 & -0.8 & -0.1 & -1 & 0.7 & 0.6 & 0 \\ -0.8 & 0.5 & -0.1 & 0.3 & -0.7 & -0.6 & -0.7 & -0.7 \\ -0.2 & -0.3 & 0.7 & -0.9 & 0 & 0 & -0.9 & 0 \\ 0.6 & 0.8 & -0.2 & 0.1 & -0.9 & -0.1 & -0.2 & 0 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.8 & -0.5 & -0.7 & 0.9 & -0.5 & 0.8 \\ 0.7 & 0.6 & -1 & 0.3 & -0.3 & 0 \\ 0.8 & -0.2 & 0.2 & 0.8 & -1 & 1 \\ 0.2 & -1 & 0.1 & 0.3 & 0.1 & -0.8 \\ 0.1 & 0.2 & -0.9 & 0.7 & 0.7 & -1 \\ 0.2 & 0.1 & 0.3 & -0.1 & -0.8 & -0.3 \\ -0.9 & -0.7 & -0.1 & 1.1 & -0.7 & -0.2 \\ -0.8 & -0.9 & -1 & -0.9 & -1 & -0.2 \\ -0.2 & 1 & -0.3 & 0.1 & 0.7 & -0.2 \\ 0.1 & -0.5 & -0.6 & 0.1 & -0.8 & -0.3 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.5 & -0.7 & 0.9 & -0.6 & -0.3 & -0.9 & 0.2 & -0.7 & -1 & 0.4 \\ -0.7 & -0.9 & 0.4 & -0.8 & 0.4 & -0.3 & -0.7 & -0.5 & -0.7 & 0.7 \\ 0.9 & -0.4 & -0.9 & -1 & 0.9 & -0.9 & -0.4 & 0.9 & 0.7 & 0 \\ -0.6 & 0.9 & 0.2 & 0.6 & -0.7 & 0.6 & -0.7 & -0.2 & 0.8 & 0 \\ 0.6 & 0.9 & -0.2 & -1 & -0.7 & 1 & -0.6 & 0 & -0.6 & -0.4 \\ 0.2 & -0.5 & -0.6 & 0.8 & -0.4 & -0.8 & -0.2 & -0.8 & 1 & -0.9 \end{bmatrix},$$

$$D = \begin{bmatrix} -0.5 & 0.4 & -1 & 0.4 & 1 & 0.3 & -0.4 & 0.3 \\ -0.6 & 0.8 & -0.2 & 0.6 & 0.3 & 0.6 & 0.3 & 0 \\ 0.8 & 0.2 & 0.3 & -0.5 & 0.1 & 0.9 & -0.3 & -0.6 \\ -0.8 & -0.7 & -0.8 & -0.6 & 0.7 & 0 & 0.9 & 0.5 \\ -1 & -0.4 & 0.5 & -0.5 & 0.6 & 0.6 & 0 & -0.9 \\ -0.3 & -0.5 & 0.9 & 0.4 & 0.3 & 0.3 & -0.8 & 0.9 \end{bmatrix}, \quad D_0 = \begin{bmatrix} 1.1 & -0.1 & -0.3 & 0.7 & 0 & -0.3 \\ 0.5 & -1.3 & -0.7 & -0.3 & -1.1 & 0.6 \\ 1.3 & 0.2 & -0.4 & 1.1 & 0.7 & 0.5 \\ -0.9 & -0.4 & 1.3 & -0.9 & -1.4 & 1.3 \\ 0.7 & 0.8 & -0.1 & -0.4 & 0.3 & -0.8 \\ -0.4 & 0.8 & -0.4 & 1.2 & 1.3 & 0.3 \end{bmatrix}.$$

It is straightforward to see that the considered model is unstable along pass as $r(A) = 3.471$ *and* $r(D_0) = 1.6$. *To compute the controllers using CLUSTER, conditions of Theorems 4.9 and 4.10 have been applied. The condition of Theorems 4.9 finished successfully and provided the feasible solution however the application of Theorem 4.10 failed. Below the limited (skipped* Z*), due to the lack of space, output of the* SDPARA *solver (after the required reformulation) is given*

$$Y = \begin{bmatrix} 66.65 & -10.01 & -25.55 & -11.06 & -7.089 & 41.31 & 5.411 & -41.61 \\ -10.01 & 45.91 & -16.63 & -26.78 & -12.45 & -45.74 & 3.741 & -3.317 \\ -25.55 & -16.63 & 30.79 & 17.12 & 16.25 & 15.5 & -4.812 & 14.48 \\ -11.06 & -26.78 & 17.12 & 28.64 & 15.38 & 20.68 & -2.95 & 16.2 \\ -7.089 & -12.45 & 16.25 & 15.38 & 22.92 & 24.52 & 0.5783 & 5.134 \\ 41.31 & -45.74 & 15.5 & 20.68 & 24.52 & 89.38 & 7.206 & -17.37 \\ 5.411 & 3.741 & -4.812 & -2.95 & 0.5783 & 7.206 & 24.5 & -2.704 \\ -41.61 & -3.317 & 14.48 & 16.2 & 5.134 & -17.37 & -2.704 & 51.89 \\ 4.024 & -16.73 & 15.33 & 7.818 & 11.46 & 29.54 & -4.55 & -1.918 \\ -14.07 & -11.52 & 14.99 & 19.91 & 11.04 & 13.38 & 2.826 & 24.6! \\ -2.784 & -9.725 & 0.5378 & 0.6203 & -8.857 & -2.837 & -0.0178 & -1.813 \\ 2.748 & -2.173 & -3.443 & -0.6931 & 3.934 & 0.3176 & -8.189 & 6.585 \\ -3.055 & -0.2784 & 2.822 & 7.731 & 4.613 & -2.564 & -4.0 & -10.07 \\ -3.207 & -0.3592 & 7.695 & -7.336 & -4.374 & 2.133 & -5.937 & -0.8617 \\ 2.06 & 8.592 & 2.949 & -2.153 & -0.4764 & 1.313 & -8.197 & -1.845 \\ -1.498 & 2.029 & 5.428 & -4.299 & -1.211 & 3.672 & 8.47 & 0.7577 \end{bmatrix}$$

$$\begin{matrix} 4.024 & -14.07 & -2.784 & 2.748 & -3.055 & -3.207 & 2.06 & -1.498 \\ -16.73 & -11.52 & -9.725 & -2.173 & -0.2784 & -0.3592 & 8.592 & 2.029 \\ 15.33 & 14.99 & 0.5378 & -3.443 & 2.822 & 7.695 & 2.949 & 5.428 \\ 7.818 & 19.91 & 0.6203 & -0.6931 & 7.731 & -7.336 & -2.153 & -4.299 \\ 11.46 & 11.04 & -8.857 & 3.934 & 4.613 & -4.374 & -0.4764 & -1.211 \\ 29.54 & 13.38 & -2.837 & 0.3176 & -2.564 & 2.133 & 1.313 & 3.672 \\ -4.55 & 2.826 & -0.0178 & -8.189 & -4.0 & -5.937 & -8.197 & 8.47 \\ -1.918 & 24.6 & -1.813 & 6.585 & -10.07 & -0.8617 & -1.845 & 0.7577 \\ 18.78 & 6.499 & 0.8356 & -1.098 & -6.499 & 5.101 & 5.696 & -3.362 \\ 6.499 & 27.27 & -3.903 & -5.707 & -1.375 & -4.909 & 7.24 & -0.1524 \\ 0.8356 & -3.903 & 44.88 & 8.118 & -1.175 & -17.72 & 20.88 & 17.36 \\ -1.098 & -5.707 & 8.118 & 48.76 & 4.138 & -12.7 & 9.748 & 17.51 \\ -6.499 & -1.375 & -1.175 & 4.138 & 31.24 & -1.084 & 0.2798 & 7.444 \\ 5.101 & -4.909 & -17.72 & -12.7 & -1.084 & 34.8 & -19.06 & -6.368 \\ 5.696 & 7.24 & 20.88 & 9.748 & 0.2798 & -19.06 & 46.01 & 16.57 \\ -3.362 & -0.1524 & 17.36 & 17.51 & 7.444 & -6.368 & 16.57 & 42.52 \end{matrix}\Bigg] ,$$

$$N = \Bigg[\begin{matrix} -19.38 & 4.428 & 8.516 & 5.313 & 5.395 & -13.24 & 3.854 & 5.563 \\ 59.01 & 46.61 & -66.71 & -38.96 & -31.93 & -48.63 & -13.58 & -39.19 \\ -44.82 & 27.07 & -6.688 & -3.486 & -17.03 & -71.37 & 7.698 & 27.6 \\ -42.3 & -12.48 & 49.88 & 16.39 & 32.01 & 36.69 & 0.6506 & 24.43 \\ -96.29 & 19.78 & 30.63 & 23.13 & 16.38 & -61.69 & 6.082 & 74.81 \\ 51.39 & -30.19 & 0.7538 & 21.0 & 28.07 & 81.04 & 14.02 & -35.31 \\ 27.53 & 31.84 & -34.15 & -41.59 & -40.85 & -45.62 & -18.65 & -21.06 \\ 75.28 & 23.86 & -69.83 & -34.04 & -34.86 & -15.71 & 5.457 & -37.45 \end{matrix}$$

$$\begin{matrix} -3.48 & 5.101 & -26.73 & -28.81 & 4.269 & 17.32 & -27.84 & -19.05 \\ -30.76 & -40.56 & -11.92 & 53.83 & 4.547 & -28.91 & 15.94 & -9.38 \\ -27.54 & 1.115 & -10.4 & -15.66 & 1.338 & 2.841 & -25.96 & -11.72 \\ 21.66 & 21.98 & -2.163 & 1.695 & 11.66 & 18.66 & 13.41 & 33.82 \\ -13.76 & 32.87 & -49.47 & -25.03 & -3.532 & 18.53 & -43.06 & -26.89 \\ 19.88 & 10.13 & -10.66 & -10.36 & 5.782 & -21.14 & 8.747 & -10.81 \\ -9.441 & -28.72 & 20.64 & 12.02 & -22.32 & 0.8115 & 25.37 & -2.809 \\ -18.48 & -28.05 & 19.74 & 17.82 & -26.09 & -39.1 & 19.35 & -20.65 \end{matrix}\Bigg] .$$

The stabilizing controllers computed due to (4.14) are given as follows

$$K_1 = \begin{bmatrix} -2.5184 & -1.6172 & -3.1514 & -0.8179 & 0.2733 & -0.3663 & -0.5098 & -2.0633 & 1.4279 & 2.0789 \\ 1.8146 & -1.2805 & 1.2826 & -1.6708 & -2.1084 & -1.2355 & 1.1579 & 1.963 & 1.8766 & -2.1932 \\ -0.7958 & -1.3533 & -0.1305 & -1.2165 & -1.2292 & -0.825 & 0.6033 & 0.0379 & 0.7917 & 0.3569 \\ 2.0925 & 4.3329 & 5.3707 & 1.941 & 0.5369 & 2.3909 & -0.5837 & 1.8155 & -4.5129 & -2.0248 \\ -2.274 & -1.7576 & -1.6418 & 0.1625 & -0.1461 & -0.7455 & 1.274 & -0.1933 & 0.8872 & -0.1612 \\ 3.9206 & 3.2178 & 5.0459 & 3.9336 & 1.0307 & 0.857 & 1.4648 & 1.8359 & -4.0747 & -3.5721 \\ -2.2454 & -2.3882 & -3.8345 & -2.8048 & -0.9492 & -0.9718 & -1.3284 & -2.2445 & 2.9953 & 3.1331 \\ 0.5058 & -1.7821 & -2.0962 & -1.5826 & 0.1563 & -1.1757 & 0.7518 & 0.1823 & 1.4603 & 0.3831 \end{bmatrix},$$

$$K_2 = \begin{bmatrix} -1.0337 & -0.3553 & -0.2657 & 0.0242 & -0.4822 & 0.9645 \\ -2.1032 & 0.6602 & 1.87 & -2.2567 & 0.9705 & -0.9634 \\ -1.0651 & -0.1905 & 0.6035 & -1.0394 & -0.2655 & -0.0107 \\ 1.9608 & 0.0777 & -0.4632 & 0.9336 & -0.3145 & -0.7875 \\ -2.0305 & -0.1257 & -0.0928 & 0.1214 & 0.7204 & 0.1113 \\ 1.2178 & 0.0699 & -0.6156 & 0.8722 & 1.1659 & -1.9622 \\ -0.8123 & 0.2242 & -0.3272 & -0.5815 & -0.2743 & 1.1007 \\ -0.3233 & 0.1759 & 0.208 & -0.685 & 0.76 & -0.581 \end{bmatrix}.$$

It is straightforward to see that the closed loop model becomes stable along the pass. To check it, the conditions of Theorem 4.5 or Theorem 4.6 or Theorem 4.7 can be applied (output matrices of that condition are skipped here due to their large dimensions). As a partial proof of the stability along the pass in the closed loop, it can be easily checked that $r(A + BK_1) = 0.49$ *and* $r(D_0 + DK_2) = 0.76$.

Remark 4.4 *It is to note that the time values given in the above tables are the time values obtained for the single solution of the considered problem. Since the cluster, on which the problem is solved, is connected with the "open" academic network using Fast Ethernet switches and it is working under TCP/IP protocol, those results depend on the current state of the network, i.e. whether other computers (not involved into cluster) in the common network use switches and hence limit the bandwidth (since the cluster members PCs are "normal" laboratory computers it was impossible to separate the cluster from the external influence, however the simulations were made during the nights, when the load of the network is relatively small).*

It is worth to mention that the counter-example, i.e. when Theorem 4.10 is successful and Theorem 4.9 fails, had not been found.

In view of this fact and above examples, it is natural to try to compare presented synthesis methods. On one hand, the application of Theorem 4.10 demands to apply much less decision variables, which is the crucial parameter (Example 4.3) and (what is the consequence) the solution is obtained much sooner. On the other hand, Theorem 4.9 appeared to be feasible in more cases than Theorem 4.10. Hence it can be supposed that Theorem 4.9 is less conservative than the other one.

This situation is somehow similar to Example 4.1, where the LMI condition involving more variables appeared to be feasible since the other has not provided the feasible solution. Hence it can be concluded that increasing the number of the decision variables causes the reduction of the condition conservativeness.

Due to the above remark, in the sequel of this dissertation, when possible (i.e. the sizes of problems are small enough), to solve the problem of synthesis (analysis) of the discrete LRP, Theorem 4.9 (Theorem 4.6) is applied.

Concluding the results of above examples, it is straightforward to see that according to the intuition, the speed of computations increases when more of PC nodes in the cluster is used. It is however to see that for the small problems, it is not sufficient to increase the number of PC members (see Table in Example 4.3, where in first 3 rows the computation times obtained with 8 PCs are smaller than those obtained with 16 PCs used in the cluster) over some limit. It is due to the fact that for the small problems and too many PCs in the cluster used, much more effort is put to communicate between nodes in the cluster and to synchronize the computations, than to take the advantage of the parallel computing. Hence the following conclusion is natural: the increase of the cluster member PCs is purposeful only for the large size problems.

The obtained results prove the usefulness of the PC clusters and SDP solvers to solve problems of analysis and synthesis of LRPs.

4.4.2 Differential case

The LMI condition similar to that given in Theorem 4.9 can be provided here for the controller design for the differential LRP (2.14)-(2.15) Firstly, define the control law over $0 \leq t < \alpha$, $k \geq 0$

$$u_{k+1}(t) = K_1 x_{k+1}(t) + K_2 y_k(t), \tag{4.16}$$

where again K_1 and K_2 are appropriately dimensioned controller matrices to be designed. Now, using the LMI stability test and defined feedback loop (4.16), the following result can be presented.

Theorem 4.11 *The differential LRP is stable along the pass under the control law of (4.16) if there exist matrices $Y > 0$, $Z > 0$, M and N of appropriate dimensions such that the following LMI holds*

$$\begin{bmatrix} YA^T + AY + N^TB^T + BN & B_0Z + BM & YC^T + N^TD^T \\ ZB_0^T + M^TB^T & -Z & ZD_0^T + M^TD^T \\ CY + DN & D_0Z + DM & -Z \end{bmatrix} < 0. \tag{4.17}$$

Then controllers K_1 and K_2 are then given by

$$K_1 = NY^{-1}, \;\; K_2 = MZ^{-1}. \tag{4.18}$$

For the proof and detailed description see [49, 139].

Example 4.5 *Consider the following unstable model of differential LRP of (2.14)-(2.15)*

$$A = \begin{bmatrix} -3.6 & 0.2 & -1.4 & -1.7 \\ 0.1 & -6.3 & 1.6 & 1.9 \\ 2.9 & -0.2 & -4.9 & -5.7 \\ -4.4 & -1.4 & -2.7 & -1.1 \end{bmatrix}, \; B = \begin{bmatrix} 1.7 & -2.2 & 1.9 \\ 1.4 & -3.4 & -3.4 \\ 1.3 & -2 & -1.6 \\ 0.5 & -2.1 & 2.8 \end{bmatrix}, \; B_0 = \begin{bmatrix} -1 & 1 \\ 0.6 & 0.6 \\ 0.5 & 0.9 \\ -0.3 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.5 & -0.4 & -0.2 & -0.1 \\ 0 & -0.6 & 0.4 & -0.6 \end{bmatrix}, \; D_0 = \begin{bmatrix} -1.4 & -0.4 \\ -0.1 & 0 \end{bmatrix}, \; D = \begin{bmatrix} -4.3 & -5.7 & -5.5 \\ 4 & 4.7 & -4.7 \end{bmatrix}.$$

The application of the LMI condition given in Theorem 4.11 provides the following matrices

$$Y = \begin{bmatrix} 1738.3753 & 1398.8361 & -84.9254 & 1393.9339 \\ 1398.8361 & 3606.6064 & 72.8413 & 2472.91 \\ -84.9254 & 72.8413 & 1141.8682 & -598.2236 \\ 1393.9339 & 2472.91 & -598.2236 & 3190.6964 \end{bmatrix}, \; Z = \begin{bmatrix} 1711.7330 & 43.6538 \\ 43.6538 & 2550.9263 \end{bmatrix},$$

$$N = \begin{bmatrix} 1553.8209 & 3379.6068 & -18.2156 & 4862.1008 \\ -1258.8102 & -2500.7549 & -51.2653 & -3718.7458 \\ -63.5197 & -596.0789 & -18.2064 & -232.6494 \end{bmatrix}, \; M = \begin{bmatrix} -182.2183 & -618.65 \\ -90.2886 & 372.7029 \\ -116.1564 & -88.4136 \end{bmatrix}.$$

Hence the controller matrices which ensure stability along the pass under the control defined in (4.16) and computed due to (4.18) become

$$K_1 = \begin{bmatrix} -0.5515 & -0.5170 & 1.2315 & 2.3964 \\ 0.3483 & 0.4909 & -1.0424 & -1.8936 \\ 0.1038 & -0.2886 & 0.0725 & 0.1190 \end{bmatrix}, \; K_2 = \begin{bmatrix} -0.1003 & -0.2408 \\ -0.0565 & 0.1471 \\ -0.0670 & -0.0335 \end{bmatrix}.$$

4.5 Determining of the 2D stability margins

In some cases, it is required to find out how far from the "stability border" is the considered discrete LRP. For the classical 1D theory (discrete case) the stability margin is defined as a maximal real number $\sigma > 0$ such that $r\big((1+\sigma)A\big) < 1$.

As for 2D discrete linear systems described by the Roesser and Fornasini Marchesini state-space models, the stability margin for discrete linear LRPs has been defined [17] as the shortest distance between a singularity of the process and the stability along the pass limit, which is the boundary of the unit bidisc, i.e. $\overline{T}^2 := \{(z_1, z_2) : |z_1| = 1, |z_2| = 1\}$. Hence the stability margin is a measure of the degree to which the process will remain stable along the pass under variations in the process state-space model matrices which define this property. Hence so-called generalized stability margin for discrete LRPs of (2.9)-(2.10) is defined as follows.

Definition 4.1 *The generalized stability margin σ_β for discrete LRPs of (2.9)-(2.10) is defined as the radius of a largest bidisc, in which the 2D characteristic polynomial satisfies*

$$\mathcal{C}(z_1, z_2) \neq 0 \,, \in \overline{U}^2_{\sigma_\beta} = \{(z_1, z_2) : |z_1| \leq 1 + (1-\beta)\,\sigma_\beta, |z_2| \leq 1 + \beta\sigma_\beta\}\,, \tag{4.19}$$

where $0 \leq \beta \leq 1$.

In particular, q_1 and q_2 give the stability margins corresponding to z_1 and z_2 respectively, i.e. along the pass and pass-to-pass respectively. In the sequel the following lemma is helpful

Lemma 4.1 *Given $q_i \in \mathbb{R}$, $q_i > 0$, $i = 1, 2$ such that*

$$\widehat{\mathcal{C}}\,(z_1, z_2) = \det \begin{bmatrix} I - z_1(1+q_1)A & -z_1(1+q_1)B_0 \\ -z_2(1+q_2)C & I - z_2(1+q_2)D_0 \end{bmatrix} \neq 0 \tag{4.20}$$

in $\overline{U}^2$, then

$$\mathcal{C}\left(z_1', z_2'\right) = \det \begin{bmatrix} I - z_1'A & -z_1'B_0 \\ -z_2'C & I - z_2'D_0 \end{bmatrix} \neq 0 \tag{4.21}$$

in U_q^2, where

$$\overline{U}_q^2 = \left\{\left(z_1', z_2'\right) : \left|z_1'\right| \leq 1 + q_1, : \left|z_2'\right| \leq 1 + q_2\right\}. \tag{4.22}$$

Now, it is possible to establish the main result on the computation of the lower bounds for the stability margins defined above.

Theorem 4.12 *For a given β such that $0 \leq \beta \leq 1$, a lower bound for the generalized stability margin σ_β is given by the solution of the following quasi-convex optimization problem:*

Maximize σ_β subject to $P > 0$, $Q > 0$, $\sigma_\beta > 0$ and the LMI

$$\begin{bmatrix} Q - P & 0 & (1+(1-\beta)\sigma_\beta)\,\widehat{A}_1^T P \\ 0 & -Q & (1+\beta\sigma_\beta)\,\widehat{A}_2^T P \\ (1+(1-\beta)\sigma_\beta)\,P\widehat{A}_1 & (1+\beta\sigma_\beta)\,P\widehat{A}_2 & -P \end{bmatrix} < 0, \tag{4.23}$$

where $\widehat{A}_2$ and $\widehat{A}_2$ are defined in (4.8).

Proof. Noting the result of Lemma 4.1 first write the condition of Theorem 4.5 for stability along the pass in terms of the matrix

$$\Upsilon = \begin{bmatrix} (1+q_1)A & (1+q_1)B_0 \\ (1+q_2)C & (1+q_2)D_0 \end{bmatrix}.$$

Now set $q_1 = (1-\beta)\sigma_\beta$ and $q_2 = \beta\sigma_\beta$, note again the definition of the stability margins, and apply the Schur complement (Lemma 3.1) with

$$\Sigma_3 = \begin{bmatrix} Q - P & 0 \\ 0 & -Q \end{bmatrix}, \quad \Sigma_1 = P^{-1}, \quad \Sigma_2 = \begin{bmatrix} (1+(1-\beta)\sigma_\beta)\,P\widehat{A}_1 & (1+\beta\sigma_\beta)\,P\widehat{A}_2 \end{bmatrix}$$

to obtain (4.23). □

In computational terms, it is straightforward to transform the maximization problem which defines the computation of the generalized eigenvalue problem which can be solved using, e.g. the MATLAB LMI CONTROL TOOLBOX. In the case of an arbitrary $\beta \in [0, 1]$, the following result can be presented.

Lemma 4.2 *For $\beta \in [0, 1]$, a lower bound for the generalized stability margins $q_1 = \beta\sigma_\beta$ and $q_2 = (1-\beta)\sigma_\beta$ can be obtained from the solution of the generalized eigenvalue problem:*

$$\textit{minimize } \lambda := \sigma_\beta^{-1},$$

$$\textit{subject to } \begin{cases} 0 < B(x), \\ A(x) < \lambda B(x), \end{cases}$$

where

$$A(x) = \begin{bmatrix} 0 & 0 & (1-\beta)\widehat{A}_1^T P \\ 0 & 0 & \beta\widehat{A}_2^T P \\ (1-\beta)P\widehat{A}_1 & \beta P\widehat{A}_2 & 0 \end{bmatrix}, \quad B(x) = -\begin{bmatrix} Q-P & 0 & \widehat{A}_1^T P \\ 0 & -Q & \widehat{A}_2^T P \\ P\widehat{A}_1^T & P\widehat{A}_2 & -P \end{bmatrix}.$$

As an example to illustrate the computation of the stability margins, the following example is provided.

Example 4.6 *Consider the discrete LRP of (2.9)-(2.10) described by the following matrices*

$$A = \begin{bmatrix} -0.35 & 0.35 & 0.32 & -0.16 \\ 0.2 & 0.09 & 0.14 & -0.21 \\ -0.12 & 0 & 0.32 & -0.16 \\ 0.36 & 0.4 & 0.16 & 0.03 \end{bmatrix}, \quad B = \begin{bmatrix} 0.45 & -0.26 \\ -0.38 & 0.41 \\ 0.68 & 0.09 \\ 0.14 & -0.11 \end{bmatrix}, \quad B_0 = \begin{bmatrix} -0.36 & -0.2 \\ -0.49 & 0.16 \\ 0.39 & -0.22 \\ -0.3 & -0.03 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.19 & 0.29 & 0.02 & -0.33 \\ 0.12 & 0.46 & 0.38 & 0.48 \end{bmatrix}, \quad D = \begin{bmatrix} -0.23 & 0.38 \\ -0.25 & 0.24 \end{bmatrix}, \quad D_0 = \begin{bmatrix} -0.44 & 0.08 \\ 0.49 & -0.08 \end{bmatrix}.$$

Beneath, the lower bounds for stability margins calculated by using the condition of Theorem 4.12 for arbitrary chosen β such that $0 \le \beta \le 1$ are presented

β	σ_β	q_1	q_2
0	0.4629	0.4629	0
0.1	0.4685	0.4216	0.0468
0.2	0.4777	0.3822	0.0955
0.3	0.481	0.3367	0.1443
0.4	0.4845	0.2907	0.1938
0.5	0.4854	0.2427	0.2427
0.6	0.4873	0.1949	0.2924
0.7	0.4823	0.1447	0.3376
0.8	0.4821	0.0964	0.3857
0.9	0.4716	0.0472	0.4244
1.0	0.4608	0	0.4608

Table 4.4. Computed values of stability margins

4.6 2D stabilization to the prescribed stability margins

The controller design objective is clearly that of achieving stability along the pass closed loop with a prescribed lower bound on the stability margins in the along the pass and pass to pass directions respectively. Here such bounds are denoted by q_1 and q_2 respectively and the following theorem regards the 2D controller design to the prescribed stability margins.

Theorem 4.13 *Discrete LRPs of the form described by (2.9)-(2.10) are stable along the pass under control laws of the form (4.11) with K defined by (4.14) and with prescribed lower bounds on the stability margins q_1, q_2, corresponding to z_1 and z_2 respectively, if there exist matrices $Y > 0$, $Z > 0$ and N of the appropriate dimensions such that the following LMI is feasible*

$$\begin{bmatrix} Z-Y & 0 & (1+q_1)\left(Y\widehat{A}_1^T + N^T\widehat{B}_1^T\right) \\ 0 & -Z & (1+q_2)\left(Y\widehat{A}_2^T + N^T\widehat{B}_2^T\right) \\ (1+q_1)\left(\widehat{A}_1Y + \widehat{B}_1N\right) & (1+q_2)\left(\widehat{A}_2Y + \widehat{B}_2N\right) & -Y \end{bmatrix} < 0. \quad (4.24)$$

Proof. For $\beta \in [0,1]$, lower bounds for the generalized stability margins are defined as follows $q_1 = \beta\sigma_\beta$, $q_2 = (1-\beta)\sigma_\beta$. Write the LMI stability along the pass condition (4.23) for the closed loop system as

$$\begin{bmatrix} Q-P & 0 & (1+q_1)\left(\widehat{A}_1^T + K^T\widehat{B}_1^T\right)P \\ 0 & -Q & (1+q_2)\left(\widehat{A}_2^T + K^T\widehat{B}_2^T\right)P \\ (1+q_1)P\left(\widehat{A}_1 + \widehat{B}_1K\right) & (1+q_2)P\left(\widehat{A}_2 + \widehat{B}_2K\right) & -P \end{bmatrix} < 0. \quad (4.25)$$

Left and right multiply (4.25) by $\mathrm{diag}(P^{-1}, P^{-1}, P^{-1})$ and substitute $Y = P^{-1}$, $Z = P^{-1}QP^{-1}$ to obtain

$$\begin{bmatrix} Z-Y & 0 & (1+q_1)\left(Y\widehat{A}_1^T + YK^T\widehat{B}_1^T\right) \\ 0 & -Z & (1+q_2)\left(Y\widehat{A}_2^T + YK^T\widehat{B}_2^T\right) \\ (1+q_1)\left(\widehat{A}_1Y + \widehat{B}_1KY\right) & (1+q_2)\left(\widehat{A}_2Y + \widehat{B}_2KY\right) & -Y \end{bmatrix} < 0.$$

Finally, use (4.14) to obtain LMI of (4.24), what completes the proof. □

Example 4.7 *To illustrate the application of Theorem 4.13 consider the following state-space model of the discrete LRP of (2.9)-(2.10) which is unstable. Assume also that the following stability margins $q_1 = 0.2$ and $q_2 = 0.3$ are required to be ensured in closed loop system under the feedback loop (4.11)*

$$A = \begin{bmatrix} -0.7 & 0.85 & 1.01 \\ -1.13 & -0.39 & -0.66 \\ -1.02 & -0.08 & 0.88 \end{bmatrix},\ B = \begin{bmatrix} 0.76 & 2.39 \\ 1.9 & -0.62 \\ -0.06 & 0.15 \end{bmatrix},\ B_0 = \begin{bmatrix} -1.16 & 0.05 \\ 0.9 & -0.75 \\ -1.15 & 0.52 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.77 & -0.19 & 0.42 \\ 0 & 0.39 & 1.11 \end{bmatrix},\ D = \begin{bmatrix} -0.75 & 0.16 \\ -0.94 & 1.14 \end{bmatrix},\ D_0 = \begin{bmatrix} -1.47 & -2.14 \\ 2.56 & 0.13 \end{bmatrix}.$$

Application of Theorem 4.13 provides the following controllers

$$K_1 = \begin{bmatrix} -0.1863 & -0.1203 & 0.4599 \\ 0.4336 & -0.2884 & -0.576 \end{bmatrix},\ K_2 = \begin{bmatrix} 0.4398 & -0.9362 \\ 0.0088 & 0.5331 \end{bmatrix},$$

which ensure the stability along the pass and drive the closed loop system to the required stability margins. The following table illustrates calculated stability margins for the closed loop system.

β	σ_β	q_1	q_2
0	0.285	0.285	0
0.1	0.2879	0.2592	0.0288
0.2	0.3447	0.2758	0.0689
0.3	0.3842	0.269	0.1153
0.4	0.4357	0.2614	0.1743
0.5	0.5081	0.2541	0.2541
0.6	0.6045	**0.2418**	**0.3627**
0.7	0.7463	**0.2239**	**0.5224**
0.8	0.9889	0.1978	0.7911
0.9	1.5122	0.1512	1.3609
1.0	2.5632	0	2.5632

Table 4.5. Obtained values of stability margins

4.7 Decoupling for the LRPs

Here, one of the "analytic" approaches to simplify the LRP synthesis tasks and hence to achieve significant numerical savings is presented [134].

When considering the discrete LRPs of (2.9)-(2.10) or (2.19)-(2.20) in some cases it is possible to simplify the structure of 2D model in the way to limit significantly the numerical efforts. It can be done by introducing the following feedback control laws (depending on the part of model which is to be simplified)

$$u_{k+1}(p) = K_y y_k(p) + \hat{u}_{k+1}(p) \tag{4.26}$$

or

$$u_{k+1}(p) = K_x x_{k+1}(p) + \hat{u}_{k+1}(p), \tag{4.27}$$

where K_x or K_y are controllers to be designed. Note that above control laws are clearly parts of (4.11) hence it can be treated that it is applied again, however K_x or K_y introduced plays a bit different role than original controllers K_1 and K_2.

The stability properties of processes described by (2.9)-(2.10) can be compactly summarized in terms of the so-called augmented plant matrix Υ (2.18) and under the action of a control law of the form (4.26) this is mapped to

$$\Upsilon_c = \begin{bmatrix} A & B_0 + BK_y \\ C & D_0 + DK_y \end{bmatrix}$$

and under the control law of the form (4.27) is mapped to

$$\Upsilon_c = \begin{bmatrix} A + BK_x & B_0 \\ C + DK_x & D_0 \end{bmatrix}.$$

For the generalized LRPs of (2.19)-(2.20) only the first of the above mappings holds.

It is straightforward to see that using (4.26), the matrix D_0 governing asymptotic stability is mapped to $D_0 + DK_y$ and it follows immediately that this control law can achieve this property in the closed loop if and only if the pair $\{D_0, D\}$ is controllable in classical 1D sense. Similarly, using (4.27), the matrix A governing the stability of every single pass is mapped to $A + BK_x$ and it follows immediately that this control law can achieve this property in the closed loop if and only if the pair $\{A, B\}$ is again controllable in the classical 1D sense.

Note also that the application of (4.26) or (4.27) influences always the second matrix, i.e. B_0 or C, hence it can be used for simplification of the model (decoupling of the dynamics).

Consider first the control law of (4.26) and assume

$$\begin{aligned} r(D_0 + DK_y) &< 1, \\ B_0 + BK_y &= 0. \end{aligned}$$

Then the state dynamics on the every current pass are in fact independent from the state dynamics from the previous passes. Hence the process becomes to some extent the series of the independent 1D systems (from the state dynamics point of view). Asymptotic stability can be achieved also, but it can be more difficult than for the case with no equality constraints, i.e. $B_0 + BK_y = 0$.

On the other hand, apply (4.27) to reach

$$\begin{aligned} r(A + BK_x) &< 1, \\ C + DK_x &= 0. \end{aligned}$$

Then from the pass profile point of view, the repetitive process dynamics are equivalent to the finite (due to that the pass length α is finite) number of independent dynamics from the pass to pass direction (for each point p, $0 \leq p \leq \alpha - 1$). Hence the asymptotic stability requirement (if satisfied) would give the same effect as the stronger along the pass one. Note that this refers only to the pass profile dynamics but $r(A + BK_x) < 1$ guarantees that the state dynamics asymptotically decreasing along the pass.

The following results can be stated on the application of control laws (4.27) or (4.26).

Theorem 4.14 *Suppose that the discrete LRP described by (2.9)-(2.10) or (2.19)-(2.20) is subject to a control law of the form (4.27). Then stability of $A + BK_x$ with simultaneous decoupling holds if and only if there exist matrices $P_x > 0$, G_x, and N_x of the appropriate dimensions such that the following LMI holds*

$$\begin{bmatrix} -P_x & AG_x + BN_x \\ G_x^T A^T + N_x^T B^T & P_x - G_x - G_x^T \end{bmatrix} < 0, \tag{4.28}$$

$$CG_x + DN_x = 0. \tag{4.29}$$

When this condition holds, K_x can be computed as

$$K_x = N_x G_x^{-1}$$

Theorem 4.15 *Suppose that the discrete LRP described by (2.9)-(2.10) is subject to a control law of the form (4.26). Then stability with decoupling holds under the action of this control law* *if and only if there exist matrices $P_y > 0$, G_y and N_y of the appropriate dimensions such that the following LMI holds*

$$\begin{bmatrix} -P_y & D_0G_y + DN_y \\ G_y^TD_0^T + N_y^TD^T & P_y - G_y - G_y^T \end{bmatrix} < 0, \tag{4.30}$$

$$B_0G_y + BN_y = 0. \tag{4.31}$$

When this condition holds, K_y can be computed as

$$K_y = N_yG_y^{-1}.$$

Proofs of both these Theorems come directly from the previous analysis together with Theorem 4.4.

It is obvious that both the results combined together give an efficient way to achieve the stability along the pass for the obtained closed loop process. Hence the following result appears.

Theorem 4.16 *Suppose that the discrete LRP described by (2.9)-(2.10) is subject to a control law of the form (4.11) (with controllers K_x and K_y). Then the both, state and the pass profile dynamics are decoupled and the overall process is stable along the pass.*

The proof is immediate on applying the results of both Theorems 4.15 and 4.14.

The approach to the controller design presented aforementioned provides a very effective means of controlling the repetitive processes considered but this is subject to several strong limitations which may hinder its applicability.

First, two matrix equations in this last result must be solved and the existence of a solution requires that

$$\text{rank}(D) = \text{rank}([D,\ CG_x])$$

and

$$\text{rank }(B) = \text{rank}([B,\ B_0G_y]).$$

The point here is that the decision matrices G_x and G_y in the LMIs are also part of the conditions for the existence of solutions which can be a source of serious problems. If no solutions exist, then one possibility is to attempt approximate decoupling by minimizing a norm applied to $CG_x + DN_x$ and $BG_y + BN_y$.

This approach can be also used for the generalized model of (2.19)-(2.20) but then the stability along the pass cannot be achieved. However, the problem of the synthesis towards asymptotic stability, which originally is related to the necessity of handling the possibly enormously large matrices (see Theorems 4.2 or related synthesis conditions) can be transformed to the significantly reduced (in the dimensionality of the problem meaning) form. In particular, suppose that the current pass state vector is decoupled from the pass profile updating equation in (2.19)-(2.20) using the control law (4.27) (i.e. (4.28)-(4.29) hold), where $\hat{u}_{k+1}(p)$ is an auxiliary current pass control input vector. Then the resulting closed loop state-space model becomes

$$x_{k+1}(p+1) = (A + BK_x)x_{k+1}(p) + B\hat{u}_{k+1}(p) + \sum_{j=0}^{\alpha-1} B_jy_k(j), \tag{4.32}$$

$$y_{k+1}(p) = D\hat{u}_{k+1}(p) + \sum_{j=0}^{\alpha-1} D_jy_k(j). \tag{4.33}$$

Hence the 1D equivalent model matrices become (according to (2.27))

$$\Phi = \begin{bmatrix} D_0 & D_1 & \dots & D_{\alpha-1} \\ D_0 & D_1 & \dots & D_{\alpha-1} \\ \vdots & \vdots & \ddots & \vdots \\ D_0 & D_1 & \dots & D_{\alpha-1} \end{bmatrix}, \ \Delta = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ 0 & D & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & D \end{bmatrix}. \tag{4.34}$$

Note that the matrix Φ of (4.34) has $(\alpha-1)m$ zero eigenvalues and the remaining m are equal to eigenvalues of the matrix $\sum_{j=0}^{\alpha-1} D_j$. Hence this fact provides in effect a computationally more efficient test for asymptotic stability for the considered case, i.e. instead of testing $r(\Phi)$, where Φ have a huge dimension, $r(\sum_{j=0}^{\alpha-1} D_j)$ can be checked. In this way, handling of large numbers of decision variables is avoided.

The form of Φ here can also be exploited in the design of a control law of the following form which is of the structure of (4.3), however it is different in such a way that is constructed from the "external" input vectors $\hat{u}_{k+1}(p)$ defined in (4.27)

$$\hat{U}(l) = KV(l), \tag{4.35}$$

where $\hat{U}(l)$ is the 1D equivalent model super-vector, to ensure asymptotic stability of the original process. In particular, suppose that the structure of the controller matrix is taken to be

$$K = \begin{bmatrix} K_0 & K_1 & \dots & K_{\alpha-1} \\ K_0 & K_1 & \dots & K_{\alpha-1} \\ \vdots & \vdots & \vdots & \vdots \\ K_0 & K_1 & \dots & K_{\alpha-1} \end{bmatrix}, \tag{4.36}$$

which yields a 1D equivalent model with state matrix

$$\widetilde{\Phi} := \Phi + \Delta K \tag{4.37}$$

and the following result can be presented.

Theorem 4.17 *Suppose that the generalized discrete LRP of the form (2.19)-(2.20) is subject to a control law (4.27), which is designed to give (4.32)-(4.33). Suppose also that (4.32)-(4.33) is expressed in 1D equivalent model form (2.24)–(2.25) (with $\hat{U}(l)$ replacing $U(l)$) and a control law of the form (4.35) (i.e. $\hat{U}(l) = KV(l)$) applied, where K has the structure of (4.36). Then the resulting closed loop process is asymptotically stable if and only if there exist matrices $P > 0$, G, N_j, $j = 0, 1, \dots, \alpha-1$ such that the following LMI holds*

$$\begin{bmatrix} -P & \sum_{j=0}^{\alpha-1}\left(D_j G + D N_j\right) \\ \sum_{j=0}^{\alpha-1}\left(G^T D_j^T + N_j^T D^T\right) & P - G - G^T \end{bmatrix} < 0. \tag{4.38}$$

Also if this condition holds, the block entry K_j in the controller matrix K of (4.36) is given by

$$K_j = N_j G^{-1}, \ \forall j = 0, 1, \dots, \alpha-1.$$

Proof. First, set $\mathcal{D} = \sum_{j=0}^{\alpha-1} D_j$, $\mathcal{K} = \sum_{j=0}^{\alpha-1} K_j$, and $\mathcal{N} = \sum_{j=0}^{\alpha-1} N_j$. In these terms, the resulting closed loop process can be recognized to be asymptotically stable if and only if there exists $W > 0$ such that

$$(\mathcal{D} + D\mathcal{K})^T W(\mathcal{D} + D\mathcal{K}) - W < 0. \tag{4.39}$$

Next, make an obvious application of the Schur complement formula to (4.39) and pre- and post-multiply the result by $\begin{bmatrix} I & 0 \\ 0 & W \end{bmatrix}$ to obtain

$$\begin{bmatrix} -W & (\mathcal{D} + D\mathcal{K})^T W \\ W(\mathcal{D} + D\mathcal{K}) & -W \end{bmatrix} < 0.$$

Now set $P = W^{-1}$ and left- and right-multiply the above LMI by diag$(P,\, P)$ to obtain

$$\begin{bmatrix} -P & P(\mathcal{D} + D\mathcal{K})^T \\ (\mathcal{D} + D\mathcal{K})P & -P \end{bmatrix} < 0.$$

Also using $\mathcal{K} = \mathcal{N}P^{-1}$ gives

$$\begin{bmatrix} -P & P\mathcal{D}^T + \mathcal{N}^T D^T \\ \mathcal{D}P + D\mathcal{N} & -P \end{bmatrix} < 0. \tag{4.40}$$

Now, it is necessary to establish that (4.40) is equivalent to (4.38) where this last condition is equivalent to

$$\begin{bmatrix} -P & \mathcal{D}G + D\mathcal{N} \\ G\mathcal{D}^T + \mathcal{N}^T D^T & P - G - G^T \end{bmatrix} < 0. \tag{4.41}$$

To show the necessity, assume that $G = P$ and note that (4.41) becomes (4.40) since the LMI is symmetric. For sufficiency, left multiply (4.41) by $[\, I \;|\mathcal{D} + D\mathcal{N}G^{-1}\,]$ (note that G is invertible since $G + G^T > 0$) and right multiply the result by its transpose to obtain (4.39). Next, note that

$$\mathcal{K} = \mathcal{N}G^{-1} = (\sum_{j=0}^{\alpha-1} N_j)G^{-1} = \sum_{j=0}^{\alpha-1}(N_j G^{-1}) = \sum_{j=0}^{\alpha-1} K_j \tag{4.42}$$

and (4.38) ensures that the controller $\mathcal{K}$ is such that

$$r\Big((\sum_{j=1}^{\alpha-1} D_j) + D\mathcal{K}\Big) < 1 \tag{4.43}$$

and, in turn (4.43) is equivalent to $r(\Phi + \Delta K) < 1$. This completes the proof. □

A simplified form of the above result arises, when it is assumed that

$$K_j = \hat{K}, \quad j = 0, 1, \ldots, \alpha - 1. \tag{4.44}$$

Also if this condition holds, then it is easy to conclude that $\mathcal{K} = \alpha\hat{K}$, and from (4.42) $\mathcal{N}$ of (4.38) satisfies $\mathcal{N} = \alpha\hat{N}$. Hence the following 1D controller matrix structure can be assumed

$$K = \begin{bmatrix} \hat{K} & \hat{K} & \ldots & \hat{K} \\ \hat{K} & \hat{K} & \ldots & \hat{K} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{K} & \hat{K} & \ldots & \hat{K} \end{bmatrix}. \tag{4.45}$$

In summary, therefore, a procedure for controller design to achieve asymptotic stability of processes described by (2.19)-(2.20) has been developed. The first step is to use preliminary feedback action to decouple the current pass state vector from the pass profile updating equation (which, however, is not always possible). This step does not reduce the dimension of the matrix Φ but crucially greatly simplifies its spectrum, which lowers the numerical efforts in the final controller design.

Example 4.8 *Consider the case of (2.19)-(2.20) when* $\alpha = 20$ *and*

$$A = \begin{bmatrix} 1.99 & -0.96 & 0 \\ 0 & 0 & -1.87 \\ -2.36 & -2.82 & -0.32 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 1.54 & 2.13 \\ 2.24 & 0 & -1.43 \\ 0.92 & -2.03 & 1.18 \end{bmatrix},$$

$$\begin{aligned}
\mathbf{B} &= \left[\, B_0 \,\middle|\, B_1 \,\middle|\, \dots \,\middle|\, B_{19} \,\right] \\
&= \left[\begin{array}{cc|cc|cc|cc|cc} -1.44 & -0.96 & -2.85 & 0 & 0 & 1.33 & 1.47 & 0 & 0.88 & -0.87 \\ 0 & -2.54 & 0 & -0.71 & 1.86 & -1.87 & 0 & 0 & 2.81 & 2.20 \\ 0 & 2.07 & 0 & 0.10 & 2.62 & 0 & 2.40 & 3.0 & 0.53 & -0.29 \end{array} \right| \\
&\quad \left| \begin{array}{cc|cc|cc|cc|cc} 0.56 & -0.09 & -2.36 & -2.63 & 0 & -2.80 & 0.74 & 2.21 & 0 & 0.03 \\ -1.88 & -0.33 & 0 & 0 & 0 & 1.92 & 1.77 & 0 & 2.58 & 0.14 \\ 1.33 & 0 & 1.0 & 0 & 0 & -1.27 & 2.83 & 0.67 & 1.04 & 0.51 \end{array} \right| \\
&\quad \left| \begin{array}{cc|cc|cc|cc|cc} 0.03 & 0 & 0 & 1.03 & -0.37 & -2.15 & 0.82 & 2.95 & -1.59 & -0.96 \\ 0 & 0.50 & 0 & 0 & 1.24 & 0.56 & 0 & 0 & 2.02 & -2.84 \\ 0 & -2.96 & 0 & -2.22 & 0 & 0 & 0 & 1.08 & -0.84 & -0.32 \end{array} \right| \\
&\quad \left| \begin{array}{cc|cc|cc|cc|cc} -2.02 & -0.47 & -1.45 & 2.09 & 0 & 0.68 & 0.25 & 0.07 & 2.60 & 0 \\ 1.94 & 0.76 & 0.60 & 0 & -0.81 & 0 & 2.87 & -1.85 & -0.15 & -0.13 \\ -2.19 & 0 & -1.18 & -1.67 & -0.98 & -2.63 & 0 & 0 & 0.99 & 0 \end{array} \right],
\end{aligned}$$

$$C = \begin{bmatrix} 1.38 & 1.69 & -2.21 \\ 0 & 0 & 1.62 \end{bmatrix}, \; D = \begin{bmatrix} -0.46 & -2.89 & 2.34 \\ 0 & -0.18 & -2.12 \end{bmatrix},$$

$$\begin{aligned}
\mathbf{D} &= \left[\, D_0 \,\middle|\, D_1 \,\middle|\, \dots \,\middle|\, D_{19} \,\right] \\
&= \left[\begin{array}{cc|cc|cc|cc|cc} 0.24 & 0 & 0 & -1.67 & 0 & 0.49 & 0 & -0.90 & -1.46 & 1.68 \\ 0.94 & 1.17 & -1.54 & 1.91 & 0 & -1.74 & -0.77 & -0.78 & 0 & -1.66 \end{array} \right| \\
&\quad \left| \begin{array}{cc|cc|cc|cc|cc} -1.92 & 0.05 & -0.6 & 0 & 0 & 1.49 & -0.16 & 0.72 & 0 & 1.13 \\ 0.95 & 0.34 & 0 & -1.79 & 0 & 0 & -0.68 & -1.43 & -1.25 & 1.87 \end{array} \right| \\
&\quad \left| \begin{array}{cc|cc|cc|cc|cc} 0 & 0 & 1.13 & -1.43 & 0 & 0 & 0 & -0.91 & 0.06 & 0.17 \\ -0.44 & 0 & 0 & 0 & 1.03 & 0 & -0.76 & 1.47 & -0.48 & 1.79 \end{array} \right| \\
&\quad \left| \begin{array}{cc|cc|cc|cc|cc} 0 & -1.15 & 0 & 0.62 & -0.04 & 0.20 & 0 & 1.45 & 0.72 & 1.49 \\ -0.83 & -1.73 & -0.77 & 0 & -0.92 & -1.75 & -1.06 & 0 & 0 & 0 \end{array} \right].
\end{aligned}$$

This model is unstable since $r(\Phi) = 6.9248 \times 10^7$. *Applying Theorem 4.14 (LMI (4.28) and the constraint (4.29)) in this case yields the solution matrices*

$$P_x = \begin{bmatrix} 1609.25 & -3472.8 & -3717.48 \\ -3472.8 & 8289.92 & 8846.9 \\ -3717.48 & 8846.9 & 9450.12 \end{bmatrix}, \; G_x = \begin{bmatrix} 970.49 & -1856.2 & -1998.97 \\ -2065.29 & 4755.64 & 5082.24 \\ -2206.71 & 5048.48 & 5413.21 \end{bmatrix},$$

$$N_x = \begin{bmatrix} -3378.43 & 65.5446 & 7105.47 \\ 108.13 & 108.64 & 90.05 \\ -1695.44 & 3848.58 & 4128.86 \end{bmatrix}$$

and the decoupling controller matrix K_x applicable in (4.27) is given by

$$K_x = \begin{bmatrix} -2.9399 & -3.7553 & 3.7526 \\ 0.8846 & 1.1064 & -0.6955 \\ -0.0751 & -0.0939 & 0.8232 \end{bmatrix}.$$

Hence the decoupling of the dynamics is granted. Then the application of Theorem 4.17 yields the following matrices (the controller structure assumptions (4.44) and (4.45) employed)

$$P = \begin{bmatrix} 407022367 & 0 \\ 0 & 407022367 \end{bmatrix}, \; G = \begin{bmatrix} 407022367 & 0 \\ 0 & 407022367 \end{bmatrix},$$

$$\hat{N} = \begin{bmatrix} -8203055 & 1106302 \\ -60008273 & 5489886 \\ -58070224 & -22833155 \end{bmatrix}$$

and hence

$$\hat{K} = \begin{bmatrix} -0.0202 & 0.0027 \\ -0.1474 & 0.0135 \\ -0.1427 & -0.0561 \end{bmatrix}.$$

Finally, it is easy to see that the resulting closed loop process is asymptotically stable as required.

Remark 4.5 *Note that the presented approach for the decoupling the dynamics can be (after the appropriate reformulations) extended for the class of differential LRPs of (2.14)-(2.15).*

4.8 Successive Stabilization Algorithm

The analysis presented in the previous section has provided a numerically efficient method of ensuring asymptotic stability of generalized discrete LRPs using the 1D equivalent model, where the numerical efforts necessary to apply it are lowered significantly. However, the requirements for the use of this approach can be difficult to met and in such a case the method may not work.

In this section, the alternative methodology of the use of the so-called successive (iterative) stabilization procedure is outlined (see [54, 70, 134]). The basic idea is that the process is driven to be asymptotically stable over a short pass length, which significantly reduces the numerical efforts necessary to accomplish the task, and then subsequently this design is augmented by increasing the pass length. It is done by using only the control action, which preserves the original repetitive process state-space model structure. This approach improves the numerical conditioning of the task to be performed and hence in many cases, allows to solve the large problems, which was not possible to do directly.

To proceed, suppose that the feedback controller matrix K applicable in (4.35) is assumed to be of the form (4.36). Such a choice for K ensures that the closed loop system retains the structure of (2.19)-(2.20). Hence K can be treated as a special set of 2D controllers K_j, $j =$

$0, \ldots, \alpha - 1$, which influence the dynamics from pass-to-pass (and along the pass as well). It can be presented in the following form

$$u_{k+1} = K_x x_{k+1}(p) + \sum_{j=0}^{\alpha-1} K_j y_k(j). \tag{4.46}$$

The structure of (4.36) for K can be achieved by setting

$$G = \begin{bmatrix} G_0 & 0 & \ldots & 0 \\ 0 & G_1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & G_{\alpha-1} \end{bmatrix} \quad \text{and} \quad N = \begin{bmatrix} N_0 & N_1 & \ldots & N_{\alpha-1} \\ N_0 & N_1 & \ldots & N_{\alpha-1} \\ \vdots & \vdots & \vdots & \vdots \\ N_0 & N_1 & \ldots & N_{\alpha-1} \end{bmatrix}$$

in the condition of Theorem 4.4. Then if this condition holds, $K_0 = N_0 G_0^{-1}$, $K_1 = N_1 G_1^{-1}$, ..., $K_{\alpha-1} = N_{\alpha-1} G_{\alpha-1}^{-1}$, and also $\widetilde{\Phi}$ can be written in the block form $\widetilde{\Phi} = [\widetilde{\Phi}_{ij}]$, where

$$\begin{aligned} \widetilde{\Phi}_{ij} &= \begin{cases} D_j + DK_j, \ i = 1, \ j = 0, 1, \ldots, \alpha - 1, \\ D_j + DK_j + \sum_{t=0}^{i-1} (\tilde{C}\tilde{A}^t B_j + \tilde{C}\tilde{A}^t BK_j), \ i = 2, 3, \ldots, \alpha, j = 0, 1, \ldots, \alpha - 1, \end{cases} \\ &= \begin{cases} \tilde{D}_j, \ i = 1, \\ \tilde{D}_j + \sum_{t=0}^{i-1} (\tilde{C}\tilde{A}^t \tilde{B}_j), \ i = 2, 3, \ldots, \alpha - 1. \end{cases} \end{aligned}$$

Hence it follows immediately that the closed loop process with the controller matrix K of the form (4.36) has a state-space model of the form (2.19)-(2.20) with $\tilde{A} = A + BK_x$, $\widetilde{B} = B$, $\tilde{C} = C + DK_x$, $\widetilde{D} = D$ and

$$\widetilde{B}_j = B_j + BK_j, \ \widetilde{D}_j = D_j + DK_j, \ j = 0, 1, \ldots, \alpha - 1.$$

Remark 4.6 *Note that the application of the controller K_x is optionally here. If the decoupling (driving C to zero) or at least the stabilization of A is required, it can be done using procedure presented in Section 4.7 (Theorem 4.14) before the successive procedure starts. If one decides that this preliminary action is not necessary (e.g. when A has been found to be Schur-stable) that part can be skipped.*

The design procedure in the framework of successive stabilization described above is as follows.

Successive Stabilization Algorithm

Step 1 Choose an initial stabilization problem size q as a multiple of m and an interval number $h > 0$. Set iteration counter $z = 1$.

Step 2 Using (4.6), (4.7) for $\Phi_{q \times q}$ compute the partial control law matrices $K_0, K_1, \ldots, K_{\frac{q}{m}-1}$.

Step 3 Modify Φ as in (4.37) using the matrices K_j, computed above under the assumption that $K_j = 0$, $\forall j \geq \frac{q}{m}$.

Step 4 Check if the resulting closed loop process is asymptotically stable. If yes, then stop and compute the resulting closed loop model matrices for the repetitive process state-space model. If no, go to *Step 5.*

Step 5 Check if $q+h > m\alpha$ and if yes, go to *Step 6.* If no, increase z by 1, set $q = q + h$ and return to *Step 2.*

Step 6 Check if $q = m\alpha$ and if yes, then terminate the algorithm since closed loop asymptotic stability cannot be achieved by this method. If no, increase z by 1, set $q = m\alpha$ and return to *Step 2.*

Remark 4.7 *For the sufficient large $m\alpha$ and small q and h the evaluation of the above algorithm can take a very long time but overall it compares very favorably with solving the stabilization problem with the full Φ.*

To highlight the evaluation of the Successive Stabilization Algorithm the following examples are given.

Example 4.9 *Consider the discrete LRP of (2.19)-(2.20) defined by the following matrices when $\alpha = 16$*

$$A = \begin{bmatrix} 0.74 & -0.03 \\ -0.02 & 0.74 \end{bmatrix},\ B = \begin{bmatrix} 1.15 \\ 1.66 \end{bmatrix},\ C = \begin{bmatrix} 0.7 & 0.34 \end{bmatrix},\ D = \begin{bmatrix} 0.8 \end{bmatrix},$$

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} B_0 \mid B_1 \mid \dots \mid B_{15} \end{bmatrix} \\ &= \left[\begin{array}{c|c|c|c|c|c|c|c} -0.45 & -0.41 & 0.78 & 0.83 & -0.38 & 0.52 & -0.41 & 0.43 \\ -0.54 & 0.11 & -0.45 & 0.56 & -0.53 & -0.14 & 0.89 & 0.57 \end{array}\right. \\ &\quad \left.\begin{array}{|c|c|c|c|c|c|c|c} -0.51 & -0.72 & 0.4 & 0.58 & -0.57 & 0.24 & 0.26 & -0.18 \\ 0.24 & -0.77 & -0.62 & -0.95 & -0.32 & -0.19 & -0.15 & 0.47 \end{array}\right], \end{aligned}$$

$$\begin{aligned} \mathbf{D} &= \begin{bmatrix} D_0 \mid D_1 \mid \dots \mid D_{15} \end{bmatrix} \\ &= \left[\begin{array}{c|c|c|c|c|c|c|c} 0.83 & -0.52 & -1.03 & 0.32 & 0.71 & 0.61 & -0.05 & 0.76 \end{array}\right. \\ &\quad \left.\begin{array}{|c|c|c|c|c|c|c|c} -0.57 & -0.29 & 1.02 & -0.07 & 0.07 & -0.98 & 0.27 & 0.37 \end{array}\right]. \end{aligned}$$

Here $r(\Phi) = 1.081$ and this process is asymptotically unstable. Attempts to design a control law towards asymptotic stability by the direct route using Theorem 4.3 or Theorem 4.4 failed. In this case it is possible to try to solve this problem applying the presented Successive Stabilization Algorithm (the preliminary stabilization of A has been skipped since it is unnecessary $r(A) < 1$).

To commence the algorithm, set $q = 2$ and $h = 1$. Then Step 1 gives

$$K_j^1 = \begin{bmatrix} -0.2546 \mid 0.3918 \end{bmatrix},$$

where the superscript denotes the iteration counter. Hence

$$\widetilde{B}_0^1 = \begin{bmatrix} -0.7468 \\ -0.9580 \end{bmatrix},\ \widetilde{B}_1^1 = \begin{bmatrix} 0.0376 \\ 0.7558 \end{bmatrix},\ \widetilde{D}_0^1 = \begin{bmatrix} 0.6236 \end{bmatrix},\ \widetilde{D}_1^1 = \begin{bmatrix} -0.2081 \end{bmatrix}$$

and $\widetilde{B}_j^1 = B_j^0$, $\widetilde{D}_j^1 = D_j^0\ \ \forall j > 1$.
Proceeding with the iterative procedure now gives in turn

$$K_j^2 = \left[\ 0.2618 \mid -0.0879 \mid 0.1173\ \right],$$

$$K_j^3 = \left[\ -0.2190 \mid 0.3807 \mid -0.0401 \mid -0.5236\ \right],$$

$$K_j^4 = \left[\ 0.2621 \mid -0.0847 \mid 0.1270 \mid -0.0042 \mid 0.0484\ \right],$$

$$K_j^5 = \left[\ -0.2002 \mid 0.3760 \mid -0.0614 \mid -0.5271 \mid 0.0150 \mid -0.3523\ \right],$$

$$K_j^6 = \left[\ 0.2771 \mid -0.0887 \mid 0.1098 \mid -0.0067 \mid 0.0605 \mid 0.0046 \mid 0.0184\ \right],$$

$$K_j^7 = \left[\ -0.1893 \mid 0.3742 \mid -0.0731 \mid -0.5292 \mid 0.0217 \mid -0.3502 \mid 0.0016 \mid -0.4778\ \right],$$

$$K_j^8 = \left[\ 0.2975 \mid -0.0973 \mid 0.0855 \mid -0.0087 \mid 0.0824 \mid 0.0145 \mid 0.0154 \mid 0.0096 \mid 0.3036\ \right],$$

$$K_j^9 = \left[\ -0.1893 \mid 0.3743 \mid -0.0732 \mid -0.5293 \mid 0.0219 \mid -0.3503 \mid 0.0017 \mid -0.4778 \mid 0 \mid 0.5242\ \right],$$

Finally, the stabilized process matrices corresponding to **B** *and* **D** *are given by*

$$\begin{aligned}\mathbf{B}^9 &= \left[\ \widetilde{B}_0 \mid \widetilde{B}_1 \mid \dots \mid \widetilde{B}_{15}\ \right] \\ &= \left[\begin{array}{c|c|c|c|c|c|c|c} -0.6719 & 0.0175 & 0.6437 & 0.2220 & -0.3593 & 0.1129 & -0.4106 & -0.1195 \\ -0.8500 & 0.7268 & -0.5668 & -0.3156 & -0.4959 & -0.7176 & 0.8909 & -0.2262 \end{array}\right. \\ &\qquad \left.\begin{array}{|c|c|c|c|c|c|c|c} -0.5133 & -0.119 & 0.4 & 0.58 & -0.57 & 0.24 & 0.26 & -0.18 \\ 0.2448 & -0.1022 & -0.62 & -0.95 & -0.32 & -0.19 & -0.15 & 0.47 \end{array}\right],\end{aligned}$$

$$\begin{aligned}\mathbf{D}^9 &= \left[\ \widetilde{D}_0 \mid \widetilde{D}_1 \mid \dots \mid \widetilde{D}_{15}\ \right] \\ &= \left[\ 0.6761 \mid -0.2221 \mid -1.0919 \mid -0.1103 \mid 0.7316 \mid 0.3313 \mid -0.0530 \mid 0.3710\right. \\ &\qquad \left.\mid -0.5662 \mid -0.1354 \mid 1.02 \mid -0.07 \mid 0.07 \mid -0.98 \mid 0.27 \mid 0.37\ \right].\end{aligned}$$

The values of $r(\widetilde{\Phi})$ *(the matrix to which* Φ *is mapped to closed loop according to (4.37)) during these iterations are given in Table 4.6 Here, the closed loop asymptotic stability is achieved after*

Iteration no.	0	1	2	3	4	5	6	7	8	9
$r(\widetilde{\Phi})$	1.09	1.18	1.7	1.13	1.92	1.38	2.02	4.34	3.5	0.86

Table 4.6. The spectral radius values of $\widetilde{\Phi}$ during the iterations

nine iterations. Note also that when the direct route was attempted, the LMI has been found to be unsolvable numerically.

Figures 4.7 a) and b) show the free evolution (i.e. $U(l) = 0, l = 0, 1, \dots$*) open and closed loop model respectively with boundary conditions* $x_k(0) = [1\ -0.5]^T$, $k = 1, 2, \dots$ *and* $y_0(p) = 0.1$, $0 \le p \le 15$.

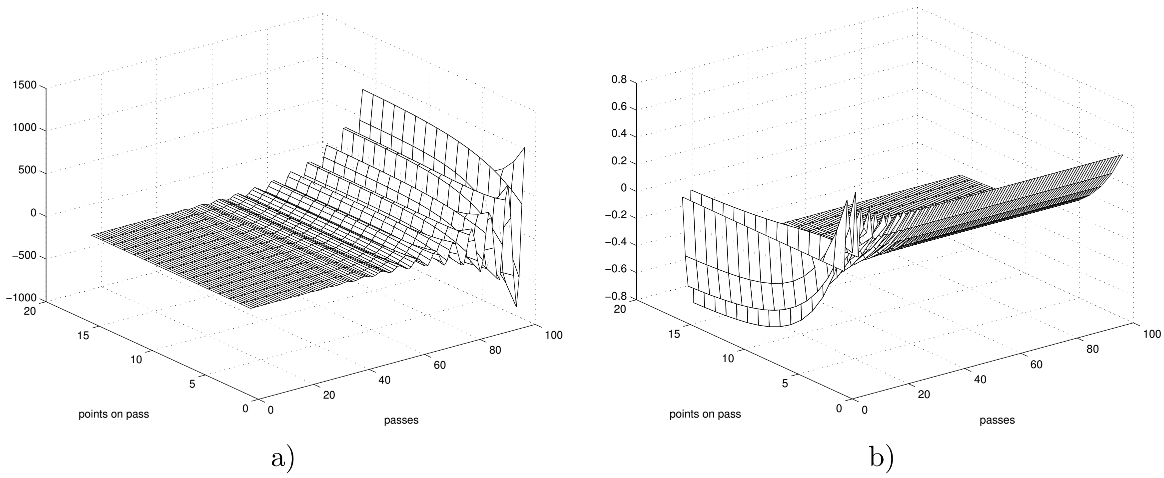


Figure 4.7. The process responses: (a)- open loop and (b)- closed loop

To prove the applicability of the successive stabilization algorithm for the cases of higher dimensioned LRPs, refer to the following example.

Example 4.10 *Consider the unstable discrete LRP of (2.19)-(2.20) defined by the following matrices when* $\alpha = 100$

$$A = \begin{bmatrix} 0.47 & 0.66 \\ 0.66 & -0.11 \end{bmatrix}, \; B = \begin{bmatrix} 0.58 \\ 0.68 \end{bmatrix},$$

$$\mathbf{B} = \left[\begin{array}{c|c|c|c|c|c|c|c|c|c} -0.81 & -0.12 & -0.2 & 0.41 & 0.03 & 0.07 & -0.2 & 0.06 & -0.14 & -0.07 \\ 0.05 & 0.04 & 0.22 & 0.4 & 0.4 & -0.35 & -0.02 & -0.2 & 0.24 & -0.03 \end{array} \right.$$

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} -0.14 & -0.15 & -0.14 & -0.15 & -0.03 & -0.06 & -0.01 & -0.01 & -0.07 & 0.02 \\ 0.02 & -0.02 & 0.02 & 0.05 & -0.05 & 0 & 0.08 & 0 & 0.02 & 0.03 \end{array}$$

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} 0.02 & -0.07 & 0.09 & -0.07 & -0.06 & 0.04 & -0.05 & -0.02 & -0.07 & -0.05 \\ 0.06 & 0.07 & 0.03 & 0.02 & 0.08 & 0.06 & 0 & -0.03 & -0.02 & -0.05 \end{array}$$

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} -0.02 & 0.04 & -0.04 & 0.06 & 0.05 & -0.01 & 0.06 & -0.02 & -0.02 & -0.05 \\ 0.02 & -0.01 & -0.02 & -0.01 & -0.04 & 0.05 & 0 & -0.03 & 0.04 & -0.03 \end{array}$$

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} -0.05 & -0.04 & -0.04 & -0.03 & 0.01 & 0.04 & 0.04 & 0.02 & 0.01 & 0.03 \\ 0.05 & 0.04 & -0.03 & 0.03 & 0.02 & 0.01 & -0.02 & -0.03 & 0.01 & -0.02 \end{array}$$

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} -0.01 & 0.04 & 0.02 & 0.02 & -0.04 & 0.01 & -0.03 & 0.03 & 0.01 & 0.03 \\ -0.02 & 0.02 & -0.03 & -0.03 & 0 & -0.02 & -0.02 & 0.03 & 0.03 & 0.03 \end{array}$$

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} 0.03 & 0.02 & -0.01 & -0.02 & 0 & -0.02 & 0 & 0.02 & 0.01 & -0.03 \\ 0 & -0.02 & 0.01 & -0.02 & -0.03 & 0 & -0.01 & -0.02 & 0.03 & 0.02 \end{array}$$

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} 0 & 0.01 & 0.03 & -0.03 & 0.01 & 0.01 & 0.01 & 0 & 0.02 & -0.02 \\ 0 & 0.01 & 0.03 & -0.01 & -0.03 & 0 & -0.02 & -0.01 & -0.01 & 0.02 \end{array}$$

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} 0 & 0.01 & -0.02 & -0.01 & 0.01 & 0 & 0.01 & -0.01 & -0.02 & 0.01 \\ -0.01 & 0.01 & -0.02 & 0 & 0.02 & -0.03 & -0.01 & 0.02 & -0.01 & -0.02 \end{array}$$

$$\left. \begin{array}{c|c|c|c|c|c|c|c|c|c} -0.01 & 0.01 & -0.01 & -0.01 & 0.01 & -0.02 & -0.01 & -0.02 & 0 & -0.01 \\ 0.02 & -0.02 & -0.02 & 0.01 & 0 & 0.02 & 0 & 0.01 & 0.01 & 0.02 \end{array} \right],$$

$$C = \begin{bmatrix} 0.16 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 0.14 \end{bmatrix},$$

$$\mathbf{D} = \big[\ -0.41 \mid -0.27 \mid -0.6 \mid 0.53 \mid 0.37 \mid -0.03 \mid 0.04 \mid 0 \mid 0.24 \mid -0.21$$
$$-0.1 \mid -0.06 \mid -0.13 \mid 0.08 \mid 0.04 \mid -0.08 \mid -0.01 \mid -0.06 \mid -0.11 \mid -0.08$$
$$0.01 \mid 0.08 \mid 0.08 \mid 0.04 \mid -0.05 \mid 0 \mid 0.05 \mid 0.03 \mid -0.08 \mid -0.03$$
$$-0.05 \mid -0.04 \mid -0.05 \mid -0.06 \mid -0.01 \mid -0.03 \mid -0.05 \mid -0.01 \mid -0.02 \mid 0.04$$
$$-0.05 \mid 0.05 \mid 0.05 \mid 0.05 \mid 0.03 \mid -0.04 \mid -0.02 \mid 0.02 \mid -0.04 \mid 0.04$$
$$0 \mid 0.04 \mid -0.01 \mid 0.03 \mid -0.04 \mid 0.01 \mid -0.01 \mid 0.01 \mid -0.02 \mid 0.02$$
$$-0.03 \mid -0.02 \mid 0 \mid 0.03 \mid -0.03 \mid 0 \mid -0.02 \mid 0.02 \mid 0 \mid 0.02$$
$$-0.02 \mid 0.02 \mid -0.03 \mid 0.01 \mid 0.03 \mid -0.01 \mid -0.01 \mid 0 \mid 0.01 \mid -0.01$$
$$0.02 \mid 0.02 \mid 0 \mid 0 \mid -0.02 \mid 0 \mid -0.02 \mid 0 \mid -0.01 \mid 0.01$$
$$0.02 \mid -0.01 \mid -0.01 \mid 0 \mid -0.01 \mid 0.02 \mid -0.01 \mid 0 \mid -0.01 \mid 0\ \big].$$

The considered LRP is unstable since $r(\Phi) = 1.4876$ and the direct application of Theorems 4.3 or 4.4 failed. Hence Successive Stabilization Algorithm has been applied with the following parameters: $q = 10$ and $h = 10$. The stability in the closed loop has been obtained after 2 iterations. The controller computed in the 2nd iteration becomes

$$K^2 = \big[\ -0.1569 \quad -0.1211 \quad -0.2795 \quad 0.2016 \quad 0.1442 \quad 0.0037 \quad 0.0277 \quad 0.0091 \quad 0.1013 \quad -0.0912$$
$$0.2275 \quad 0.2096 \quad 0.2622 \quad 0.0135 \quad 0.0045 \quad 0.1447 \quad -0.0190 \quad 0.0781 \quad 0.1782 \quad 0.0604\ \big]$$

and it is easy to check that application of this controller provides the considered model in the closed loop configuration asymptotically stable.

Remark 4.8 *There arises the natural question of the applications of the presented above algorithm in comparison with application of PC clusters in problems of stabilization of LRPs. Note that those solutions for dealing with possible large dimensions problems are not contradictory. Hence those ways can be applied interchangeably. It can be said that solutions like the above algorithm try to simplify the problem itself, exploiting its characteristic structure and PC clusters provide the unlimited (in theory) computational capability, which can help to overcome all problems which appear when the high computational ability is required.*

What is more, there are no obstacles to use those two approaches together, i.e. apply the successive stabilization algorithm run on PC cluster. In that point of view the successive stabilization algorithm can be treated as a preprocessing phase to obtain the easier problem from the computational viewpoint, and the usage of the supercomputer allows to solve this task much faster than the original one (large dimensioned).

4.9 Output feedback based controller design

The previous considerations regarding the synthesis (the controller design) of LRP towards stability along the pass were based on the assumption that for the control issues, the full information from the past, i.e. state vectors and pass profiles, were available for use. In some cases the state vector $x_{k+1}(p)$ may not be available or, at best, only some of its entries are. In this situation two approaches for the stabilization are available. First possibility to deal with this is the construction of the 2D state observer and the application of that estimated state during the controller design procedure (see e.g. [140, 141, 142]).

The second way is to try to control the LRP directly, using only the pass profile (outputs) vectors. This approach has been outlined in [135] or with extended control law in [137, 136] and is the original author's contribution.

The application of the output based feedback control laws to achieve closed loop stability along the pass is now considered. The first control law considered here has the following form over $0 \le p \le \alpha - 1,\ k \ge 0$

$$u_{k+1}(p) = \widetilde{K}_1 y_{k+1}(p) + \widetilde{K}_2 y_k(p). \tag{4.47}$$

This control law is, in general, weaker than that of (4.11) and examples are easily given where stability along the pass can be achieved using (4.11) but not (4.47). It is important to note here that by definition the pass profile produced on each pass is available for control purposes before the start of each new pass. As such, this control law (and extensions) assumes storage of the required previous pass profiles and that they are not corrupted by noise etc.

The block scheme of the closed loop process under the control law of (4.47) is shown in Figure 4.8.

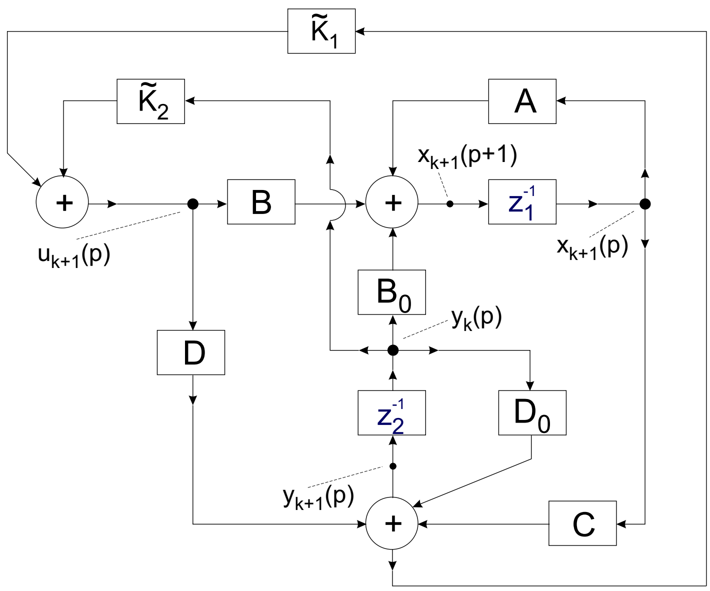


Figure 4.8. Output control scheme

To consider the effect of a controller of the form (4.47) on the process dynamics, first substitute the pass profile equation of (2.10) into (4.47) to obtain (assuming the required matrix inverse exists)

$$u_{k+1}(p) = (I - \widetilde{K}_1 D)^{-1} \widetilde{K}_1 C x_{k+1}(p) + (I - \widetilde{K}_1 D)^{-1} [\widetilde{K}_2 + \widetilde{K}_1 D_0]\, y_k(p) \tag{4.48}$$

and hence (4.48) can be treated as a particular case of (4.11) with

$$\begin{aligned} K_1 &= (I - \widetilde{K}_1 D)^{-1} \widetilde{K}_1 C, \\ K_2 &= (I - \widetilde{K}_1 D)^{-1} (\widetilde{K}_2 + \widetilde{K}_1 D_0). \end{aligned} \tag{4.49}$$

This route may encounter serious numerical difficulties (arising from the fact that (4.49) is a set of matrix nonlinear algebraic equations) and hence proceed with it by rewriting these last equations to obtain

$$\begin{aligned}(I - \widetilde{K}_1 D)K_1 &= \widetilde{K}_1 C,\\ (I - \widetilde{K}_1 D)K_2 &= \widetilde{K}_2 + \widetilde{K}_1 D_0,\end{aligned}$$

and assume that

$$K_1 = L_1 C. \tag{4.50}$$

Note that this assumption imposes no restrictions on the results developed here but could be a source of difficulty in other cases e.g. in uncertainty analysis, where the resulting robust control problem may not be convex.

It now follows immediately that

$$\begin{aligned}\widetilde{K}_1 &= L_1(I + DL_1)^{-1}, \qquad (4.51)\\ \widetilde{K}_2 &= [I - L_1(I + DL_1)^{-1}D]K_2 - L_1(I + DL_1)^{-1}D_0,\end{aligned}$$

for any L_1 such that $I + DL_1$ is nonsingular, and the following result can be presented.

Theorem 4.18 *Suppose that the discrete LRP of the form described by (2.9)-(2.10) is subject to a control law of the form (4.47) and that (4.50) holds. Then the resulting closed loop process is stable along the pass if there exist matrices $Y > 0,\ Z > 0, X > 0$ and N such that the following LMI holds*

$$\begin{bmatrix} Z - Y & (*) & (*) \\ 0 & -Z & (*) \\ \widehat{A}_1 Y + \widehat{B}_1 N\widetilde{C} & \widehat{A}_2 Y + \widehat{B}_2 N\widetilde{C} & -Y \end{bmatrix} < 0, \tag{4.52}$$

$$X\widetilde{C} = \widetilde{C}Y,$$

where $\widehat{B}_1,\ \widehat{B}_2,\ \widehat{A}_1,\ \widehat{A}_2, N$ are defined as in Theorem 4.9, and $\widetilde{C} = diag(C, I)$. Also if this condition holds, the controller matrices $\widetilde{K}_1$ and $\widetilde{K}_2$ can be obtained from (4.51), where

$$\begin{bmatrix} L_1 & K_2 \end{bmatrix} = NX^{-1} \tag{4.53}$$

and it is required that $I + DL_1$ is nonsingular.

Proof. From (4.53) it is straightforward to see that $N = LX$, $L := [L_1\ \ K_2]$ and substitution into the LMI of (4.52) now gives with $X\widetilde{C} = \widetilde{C}Y$ applied

$$\begin{bmatrix} Z - Y & (*) & (*) \\ 0 & -Z & (*) \\ \widehat{A}_1 Y + \widehat{B}_1 L\widetilde{C}Y & \widehat{A}_2 Y + \widehat{B}_2 L\widetilde{C}Y & -Y \end{bmatrix} < 0.$$

Finally, set $L\widetilde{C} = K$ to obtain the LMI stabilization condition (i.e Theorem 4.5 or 4.6 applied to the closed loop process), which completes the proof. □

The design developed above is easily implemented using LMI toolboxes such as SCILAB LMI OPTIMIZATION PACKAGE or MATLAB LMI CONTROL TOOLBOX, but has the possible disadvantage that it is based on a sufficient but not necessary stability condition. (Also the equality constraint of (4.50) can be a source of the serious numerical difficulties when using the MATLAB but thanks to the application of SCILAB such problems are avoided and hence SCILAB is used in the numerical computations reported here.) This means that there could appear not insignificant degree of conservativeness in the sense that in some cases it will fails to produce a stabilizing controller, when one actually exists. To decrease the level of conservativeness present, an extension of the control law considered in this section based on the additional use of the delayed current pass profile and pass-to-pass profile information, is developed. It is to note that the pass profile has already been generated by the process evolution and hence is available for the control purposes.

Example 4.11 *Consider the case of (2.9)-(2.10) defined by*

$$A = \begin{bmatrix} 0.06 & -1.62 & 0 \\ -0.98 & 0.28 & -2.89 \\ 0.03 & 2.66 & 2.63 \end{bmatrix},\ B = \begin{bmatrix} -1.43 & -2.13 \\ 1.23 & 1.48 \\ 2.91 & -2.18 \end{bmatrix},\ B_0 = \begin{bmatrix} 0.04 & 0 \\ 0 & 0 \\ 0 & 1.04 \end{bmatrix},$$

$$C = \begin{bmatrix} -1.40 & -0.03 & -2.70 \\ 0.52 & 0 & -2.15 \end{bmatrix},\ D = \begin{bmatrix} -1.64 & -0.52 \\ -0.71 & 0.11 \end{bmatrix},\ D_0 = \begin{bmatrix} -0.28 & -0.31 \\ 1.15 & -0.31 \end{bmatrix}.$$

In this case, the design algorithm of Theorem 4.18 is successful with $X = diag(X_1, X_2)$, where

$$X_1 = \begin{bmatrix} 606209.9 & -862471.7 \\ -862471.7 & 1346539.1 \end{bmatrix},\ X_2 = \begin{bmatrix} 5077595.6 & -2067002.5 \\ -2067002.5 & 11684609.3 \end{bmatrix}$$

$$\text{and } N = \begin{bmatrix} -178963.77 & 260104.7 & 439963.13 & -2530866.6 \\ 4116.45 & -35496.08 & -360858.98 & 2093081.41 \end{bmatrix},$$

where Y and Z are omitted due to space limitations. Then the matrices L_1 and K_2 of (4.53) are

$$L_1 = \begin{bmatrix} -0.2299 & 0.0459 \\ -0.3462 & -0.2481 \end{bmatrix},\ K_2 = \begin{bmatrix} -0.0016 & -0.2169 \\ 0.002 & 0.1795 \end{bmatrix},$$

which using (4.51) yield the output controller matrices applicable in (4.47)

$$\widetilde{K}_1 = \begin{bmatrix} -0.1523 & 0.0576 \\ -0.2020 & -0.2524 \end{bmatrix},\ \widetilde{K}_2 = \begin{bmatrix} -0.1103 & -0.2163 \\ 0.2363 & 0.1355 \end{bmatrix}.$$

4.9.1 Output controller design (extension)

In Section 4.9 the control law, which included a contribution from the last but one pass profile has been used. Here, the investigation of the application of delayed current pass profile information in the control law is done. First define the delay operators z_1, z_2 in the along the pass (p) and pass-to-pass (k) directions, respectively, as (2.13)

$$x_k(p) := z_1 x_k(p+1),\ \ x_k(p) := z_2 x_{k+1}(p).$$

The particular control law investigated here is given by

$$u_{k+1}(p) = \widetilde{K}_1 y_{k+1}(p) + \widetilde{K}_2 y_k(p) + \widetilde{K}_3 y_{k+1}(p-1), \tag{4.54}$$

which can be recasted to

$$u_{k+1}(p) = K_1 x_{k+1}(p) + K_2 y_k(p) + K_3 y_{k+1}(p-1), \qquad (4.55)$$

where

$$\begin{aligned} K_1 &= (I - \widetilde{K}_1 D)^{-1} \widetilde{K}_1 C, \\ K_2 &= (I - \widetilde{K}_1 D)^{-1} (\widetilde{K}_2 + \widetilde{K}_1 D_0), \\ K_3 &= (I - \widetilde{K}_1 D)^{-1} \widetilde{K}_3. \end{aligned}$$

Again, by assumption that $K_1 = L_1 C$ it is to see that

$$\begin{aligned} \widetilde{K}_1 &= L_1 (I + DL_1)^{-1}, \\ \widetilde{K}_2 &= [I - L_1 (I + DL_1)^{-1} D] K_2 - L_1 (I + DL_1)^{-1} D_0, \\ \widetilde{K}_3 &= [I - L_1 (I + DL_1)^{-1} D] K_3. \end{aligned}$$

The closed loop process now is given by

$$\begin{aligned} x_{k+1}(p+1) &= (A + BL_1 C) x_{k+1}(p) + (B_0 + BK_2) y_k(p) + BK_3 y_{k+1}(p-1), \\ y_{k+1}(p) &= (C + DL_1 C) x_{k+1}(p) + (D_0 + DK_2) y_k(p) + DK_3 y_{k+1}(p-1). \end{aligned}$$

This last description is again not in the form, to which Theorem 4.5 can be applied but, by the following method presented in Section 4.9, it is possible to obtain an equivalent state-space model, for which this is the case. In particular, apply (2.13) to the above model to obtain

$$\begin{aligned} x_k(p) &= z_1 (A + BL_1 C) x_k(p) + z_1 z_2 (B_0 + BK_2) y_k(p) + z_1^2 BK_3 y_k(p), \\ y_k(p) &= (C + DL_1 C) x_k(p) + z_2 (D_0 + DK_2) y_k(p) + z_1^2 DK_3 y_k(p) \end{aligned}$$

and introduce the following characteristic polynomial of the above system

$$\mathcal{C}_c(z_1, z_2) := \det \begin{bmatrix} I - z_1 \widetilde{A} & -z_1 z_2 \widetilde{B}_0 - z_1^2 BK_3 \\ -\widetilde{C} & I - z_2 \widetilde{D}_0 - z_1^2 DK_3 \end{bmatrix},$$

which is obviously equivalent to replacing the right-hand side by

$$\det \begin{bmatrix} I - z_1 \widetilde{A} & -z_2 \widetilde{B}_0 - z_1 BK_3 \\ -z_1 \widetilde{C} & I - z_2 \widetilde{D}_0 - z_1^2 DK_3 \end{bmatrix}.$$

The application of appropriate elementary operations (which leave the determinant invariant) to the right-hand side of this last expression now yields that it can be replaced by

$$\mathcal{C}_c(z_1, z_2) = \det \begin{bmatrix} I - z_1 \widetilde{A} & 0 & -z_2 \widetilde{B}_0 - z_1 BK_3 \\ 0 & I & -z_1 DK_3 \\ -z_1 \widetilde{C} & -z_1 I & I - z_2 \widetilde{D}_0 \end{bmatrix}. \qquad (4.56)$$

where

$$\begin{aligned} \widetilde{A} = A + BL_1 C,\ \widetilde{B}_0 = B_0 + BK_2, \\ \widetilde{C} = C + DL_1 C,\ \widetilde{D}_0 = D_0 + DK_2 \end{aligned}$$

and define

$$\widetilde{A}_1 = \begin{bmatrix} \widetilde{A} & 0 & BK_3 \\ 0 & 0 & DK_3 \\ \widetilde{C} & I & 0 \end{bmatrix}, \ \widetilde{A}_2 = \begin{bmatrix} 0 & 0 & \widetilde{B}_0 \\ 0 & 0 & 0 \\ 0 & 0 & \widetilde{D}_0 \end{bmatrix}. \tag{4.57}$$

Now, it is possible to replace the right-hand side of the expression defining $\mathcal{C}_c(z_1, z_2)$ of (4.56) by

$$\det(I - z_1\widetilde{A}_1 - z_2\widetilde{A}_2), \tag{4.58}$$

where

$$\widetilde{A}_1 = A_1 + B_1K, \quad \widetilde{A}_2 = A_2 + B_2K$$

and

$$A_1 = \begin{bmatrix} A & 0 & 0 \\ 0 & 0 & 0 \\ C & I & 0 \end{bmatrix}, \ A_2 = \begin{bmatrix} 0 & 0 & B_0 \\ 0 & 0 & 0 \\ 0 & 0 & D_0 \end{bmatrix}, \ B_1 = \begin{bmatrix} B & 0 & B \\ 0 & 0 & D \\ D & 0 & 0 \end{bmatrix}, \ B_2 = \begin{bmatrix} 0 & B & 0 \\ 0 & 0 & 0 \\ 0 & D & 0 \end{bmatrix} \tag{4.59}$$

$$\text{and finally } K = \begin{bmatrix} K_1 & 0 & 0 \\ 0 & 0 & K_2 \\ 0 & 0 & K_3 \end{bmatrix}. \tag{4.60}$$

Remark 4.9 *Note that the characteristic polynomial (4.58) is the same as the characteristic polynomial of the 2D FM and it has the same form as the characteristic polynomial of (2.12), defined for the discrete LRP. Hence known procedures for the controller design can be applied.*

Due to the above remark, Theorem 4.18 is now applicable and the following result can be presented.

Theorem 4.19 *Suppose that a discrete LRP of (2.9)-(2.10) is subject to a control law defined by (4.55) with K_1 satisfying (4.50). Then the resulting closed loop process is stable along the pass if there exist matrices $Y > 0$, $X = diag(X_1, X_2, X_3) > 0$ and $Z > 0$ such that*

$$\begin{bmatrix} Z - Y & (*) & (*) \\ 0 & -Z & (*) \\ A_1Y + B_1N\widehat{C} & A_2Y + B_2N\widehat{C} & -Y \end{bmatrix} < 0, \tag{4.61}$$

$$X\widehat{C} = \widehat{C}Y,$$

where A_1, A_2, B_1, B_2 are given by (4.59),

$$\widehat{C} = \begin{bmatrix} C & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \ N = \begin{bmatrix} N_1 & 0 & 0 \\ 0 & 0 & N_2 \\ 0 & 0 & N_3 \end{bmatrix} \tag{4.62}$$

$$\textit{and} \ \begin{bmatrix} L_1 & 0 & 0 \\ 0 & 0 & K_2 \\ 0 & 0 & K_3 \end{bmatrix} = NX^{-1}. \tag{4.63}$$

Proof. This is virtually identical to that of Theorem 4.18 and hence the details are omitted here. □

4.9.2 Output controller design (further extensions)

In this section another set of additional delayed factors joined to the control law is considered. The control law considered in this section has the following form and is, in effect, (4.47) augmented at the point p by additive contributions from the same point on the last but one pass profile and the point $p-1$ on the previous pass profile

$$u_{k+1}(p) = \widetilde{K}_1 y_{k+1}(p) + \widetilde{K}_2 y_k(p) + \widetilde{K}_3 y_k(p-1) + \widetilde{K}_4 y_{k-1}(p). \tag{4.64}$$

Substituting (2.10) into the control law (4.64) now yields that this last control law is, in fact, a particular case of the so-called extended, mixed state, pass profile controller

$$u_{k+1}(p) = K_1 x_{k+1}(p) + K_2 y_k(p) + K_3 y_k(p-1) + K_4 y_{k-1}(p). \tag{4.65}$$

This last control law is, in effect, again an extension of (4.47) but here it is used as an intermediate step in the computation of the matrices $\widetilde{K}_i,\ i = 1, \ldots, 4$, through use of the following result.

Theorem 4.20 *Suppose that the discrete LRP of the form described by (2.9)-(2.10) is subject to a control law of the form (4.65) and that (4.50) holds. Then the resulting closed loop process is stable along the pass if there exist matrices* $Y > 0$, $X = diag(X_1, X_2, X_3, X_4) > 0$, $Z > 0$ *and* N *such that*

$$\begin{bmatrix} Z-Y & (*) & (*) \\ 0 & -Z & (*) \\ \widehat{A}_1 Y + \widehat{B}_1 N \widehat{C} & \widehat{A}_2 Y + \widehat{B}_2 N \widehat{C} & -Y \end{bmatrix} < 0, \tag{4.66}$$
$$X\widehat{C} = \widehat{C}Y,$$

where

$$\widehat{A}_1 = \begin{bmatrix} A & -I & 0 & B_0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\ \widehat{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ C & 0 & -I & D_0 \end{bmatrix},\ \widehat{B}_1 = \begin{bmatrix} B & 0 & 0 & B \\ 0 & B & 0 & 0 \\ 0 & D & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\widehat{B}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & B & 0 \\ 0 & 0 & D & 0 \\ D & 0 & 0 & D \end{bmatrix},\ N = \begin{bmatrix} N_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -N_3 \\ 0 & 0 & 0 & -N_4 \\ 0 & 0 & 0 & N_2 \end{bmatrix},\ \widehat{C} = \begin{bmatrix} C & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix},$$

with

$$\begin{bmatrix} L_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_3 \\ 0 & 0 & 0 & K_4 \\ 0 & 0 & 0 & K_2 \end{bmatrix} = NX^{-1}. \tag{4.67}$$

Also if (4.66) holds, the controller matrices $\widetilde{K}_1$ *and* $\widetilde{K}_2$ *can be computed using (4.51) and then*

$$\begin{aligned} \widetilde{K}_3 &= [I - L_1(I + DL_1)^{-1}D]K_3, \\ \widetilde{K}_4 &= [I - L_1(I + DL_1)^{-1}D]K_4, \end{aligned} \tag{4.68}$$

where it is assumed that $I + DL_1$ *is nonsingular.*

Proof. Substitute (4.65) into (2.9)-(2.10) and using (4.50) the following closed loop state-space model is obtained

$$\begin{aligned} x_{k+1}(p+1) &= (A+BL_1C)x_{k+1}(p)+(B_0+BK_2)y_k(p)+BK_3y_k(p-1)+BK_4y_{k-1}(p), \\ y_{k+1}(p) &= (C+DL_1C)x_{k+1}(p)+(D_0+DK_2)y_k(p)+DK_3y_k(p-1)+DK_4y_{k-1}(p). \end{aligned} \quad (4.69)$$

This last description is not in the form to which Theorem 4.5 can be applied but it is possible to obtain an equivalent state-space model for which this is the case. Here the route is by using the delay operators of (2.13) and the 2D characteristic polynomial. To begin, apply (2.13) to (4.69) to obtain the closed loop model. Next, again introduce the characteristic polynomial of that as

$$\mathcal{C}_c(z_1,z_2) := det\begin{bmatrix} I-z_1\widetilde{A} & -z_1\widetilde{B}_0 - z_1^2F_1 - z_1z_2F_3 \\ -z_2\widetilde{C} & I - z_2\widetilde{D}_0 - z_1z_2F_2 - z_2^2F_4 \end{bmatrix},$$

$$\begin{aligned} \text{where } \widetilde{A} = A + BL_1C,\ \widetilde{B}_0 = B_0 + BK_2,\ F_1 = BK_3,\ F_2 = DK_3, \\ \widetilde{C} = C + DL_1C,\ \widetilde{D}_0 = D_0 + DK_2,\ F_3 = BK_4,\ F_4 = DK_4. \end{aligned}$$

The application of the appropriate elementary operations (which leave the determinant invariant) to the right-hand side of this last expression now yields that it can be replaced by

$$det\begin{bmatrix} I-z_1\widetilde{A} & z_1I & 0 & -z_1\widetilde{B}_0 \\ 0 & I & 0 & z_1F_1+z_2F_3 \\ 0 & 0 & I & z_1F_2+z_2F_4 \\ -z_2\widetilde{C} & 0 & z_2I & I-z_2\widetilde{D}_0 \end{bmatrix}. \quad (4.70)$$

At this stage, the closed loop state-space model has a 2D characteristic polynomial, which is of the form required for use (in the form of (4.58)) and therefore Theorem 4.18 can be directly applied.

Application of Theorem 4.5 together with some algebraic operations now yield directly the LMI of (4.66) as a sufficient condition for the closed loop stability along the pass. Finally, by an identical argument to that of the previous section, it is straightforward to see that $\widetilde{K}_1$ and $\widetilde{K}_2$ can be computed using (4.51) and $\widetilde{K}_3$ and $\widetilde{K}_4$ using (4.68), provided that $I+DL_1$ is nonsingular, and the proof is complete. □

Example 4.12 *As a numerical example consider the following process, with* $x_{k+1}(0) = 1,\ k \geq 0,\ y_0(p) = 1,\ 1 \leq p \leq 19$, *which is unstable along the pass since* $r(D_0) > 1$,

$$A = \begin{bmatrix} -1.36 & -1.29 & -0.8 \\ 0.15 & 0.34 & 0 \\ -0.19 & 0 & -1.36 \end{bmatrix},\ B_0 = \begin{bmatrix} 0.44 & 0.51 \\ 0.93 & 0.14 \\ 0.65 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0.18 & -2.35 & 0.8 \\ 1.07 & -2.5 & 0.5 \\ -0.43 & 0.8 & 2.82 \end{bmatrix},$$

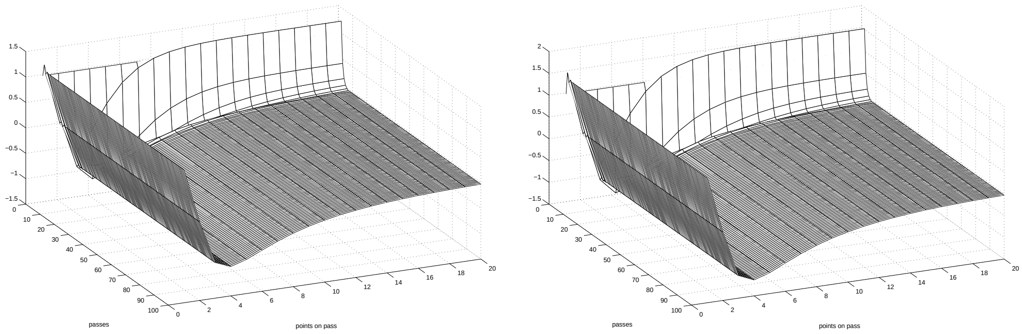
$$C = \begin{bmatrix} -0.38 & 0 & -0.37 \\ 0 & 0 & -0.98 \end{bmatrix},\ D = \begin{bmatrix} -2.85 & -0.65 & -2.5 \\ -0.28 & -2.98 & 1.96 \end{bmatrix},\ D_0 = \begin{bmatrix} -1.15 & 0 \\ -0.42 & 1.13 \end{bmatrix}.$$

In this case

$$\widetilde{K}_1 = \begin{bmatrix} 49.5 & -40.8 \\ 14.27 & -11.77 \\ -44.49 & 36.46 \end{bmatrix},\ \widetilde{K}_2 = \begin{bmatrix} -1.77 & 0.96 \\ -0.18 & 0.31 \\ 1.26 & -0.97 \end{bmatrix},$$

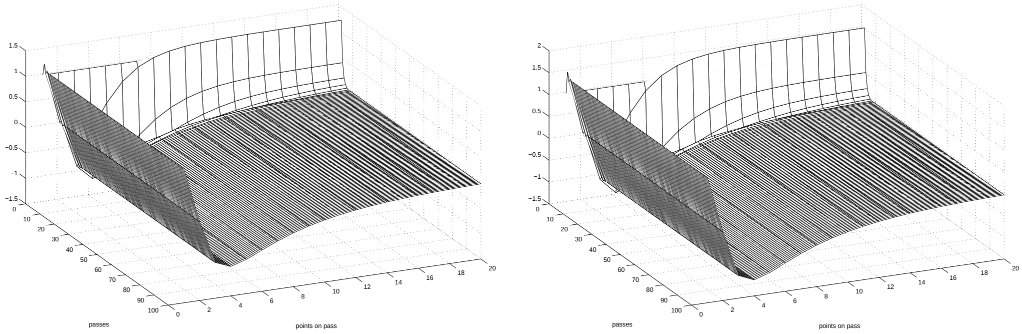
$$\widetilde{K}_3 = 10^{-12} \times \begin{bmatrix} 0.37 & -0.33 \\ 0.11 & -0.1 \\ -0.33 & 0.29 \end{bmatrix}, \ \widetilde{K}_4 = 10^{-13} \times \begin{bmatrix} 0.68 & -0.6 \\ 0.2 & -0.17 \\ -0.61 & 0.53 \end{bmatrix}.$$

Figure 4.9 shows the corresponding stable along the pass responses with the control law of (4.47) defined in Section 4.9 applied. Figure 4.10 shows the stable along the pass responses with the control law of (4.64) applied. It is straightforward to notice that responses of the controlled processes with partial or full set of controllers do not differ.



Channel 1 Channel 2

Figure 4.9. The controlled responses (only controllers $\widetilde{K}_1$ and $\widetilde{K}_2$ applied)



Channel 1 Channel 2

Figure 4.10. The controlled responses (full set of controllers applied)

Remark 4.10 *Note that in the example here the elements of $\widetilde{K}_3$ and $\widetilde{K}_4$ are significantly smaller in magnitude than those in the other controller matrices. Also if these matrices are deleted from the control law then it can be verified that the closed loop process is still stable along the pass and there is very little difference in the controlled response. Note also that direct use of the design method of Theorem 4.18 fails to give a stable design. Hence it can be conjectured that this last design method can be exploited to reduce the degree of conservativeness due to the use of a sufficient but not necessary stability condition.*

The presented method of improving the control law by the adding the delayed factors proved its applicability. Nevertheless, here only two possible choices of the delayed factors incorporated into the control law (and the way how to deal with them) have been presented. Naturally, those particular choices have not exploited the whole palette of abilities for the right choice of the delayed factors used in the control law. It is straightforward to see that there can be proposed any other structure of the control law and it can provide more conservativeness of the stability condition reduction. What is worth mentioning here that presented stability conditions are the sufficient ones and by increasing the number of the additional factors used in the control law, it can be treated as approaching to the sufficient and necessary condition (the similar work on LMI sufficient and necessary stability conditions obtained by increasing the state vector by additional delayed factors for 2D systems can be found in [52]).

However, it has be noted that, in theory there can be used any of the additional set of delayed factors to construct the control law, but on the other hand, in practice, the resulting LMI can be so high dimensioned that it can be unsolvable using the single PC computer e.g. due to the RAM limits (for comparison see Example 2.1). Hence as a middle-range remedy, the PC cluster can be used for computations. Yet, it has to be outlined that from the practical viewpoint, the unlimited increasing the number of additional factors in the control law is impossible.

4.9.3 Output controller design for differential LRPs

It is also possible to define the output control scheme for the differential LRP of (2.14)-(2.15). Again it is assumed that the state vector $x_{k+1}(t)$ may not be available or, at best, only some of its entries are. Hence the use of output based feedback based control laws to achieve closed loop stability along the pass is assumed. The control law has the following form over $0 \leq t < \alpha$, $k \geq 0$

$$u_{k+1}(t) = \widetilde{K}_1 y_{k+1}(t) + \widetilde{K}_2 y_k(t). \tag{4.71}$$

This control law is, in general, weaker than that of (4.16) and examples are easily given where stability along the pass can be achieved using (4.16) but not (4.71). It is important to note here that by definition the pass profile produced on each pass is available for control purposes before the start of each new pass. As such, this control law (and extensions) assumes storage of the required previous pass profiles and that they are not corrupted by noise etc.

To consider the effect of a controller of the form (4.71) on the process dynamics, first substitute the pass profile equation of (2.15) into (4.71) to obtain (assuming the required matrix inverse exists)

$$u_{k+1}(t) = (I - \widetilde{K}_1 D)^{-1} \widetilde{K}_1 C x_{k+1}(t) + (I - \widetilde{K}_1 D)^{-1} [\widetilde{K}_2 + \widetilde{K}_1 D_0] y_k(t) \tag{4.72}$$

and hence (4.72) can be treated as a particular case of (4.16) with

$$\begin{aligned} K_1 &= (I - \widetilde{K}_1 D)^{-1} \widetilde{K}_1 C, \\ K_2 &= (I - \widetilde{K}_1 D)^{-1} (\widetilde{K}_2 + \widetilde{K}_1 D_0). \end{aligned} \tag{4.73}$$

This route may encounter serious numerical difficulties (arising from the fact that (4.73) is a set of matrix nonlinear algebraic equations) and hence it is possible to proceed by rewriting these last equations finally to obtain

$$\begin{aligned} (I - \widetilde{K}_1 D) K_1 &= \widetilde{K}_1 C, \\ (I - \widetilde{K}_1 D) K_2 &= \widetilde{K}_2 + \widetilde{K}_1 D_0 \end{aligned} \tag{4.74}$$

and assume that

$$K_1 = L_1 C. \tag{4.75}$$

Note that this assumption imposes no restrictions on the results developed here but could be a source of difficulty in other cases, e.g. in uncertainty analysis where the resulting robust control problem may not be convex.

It now follows immediately that

$$\begin{aligned} \widetilde{K}_1 &= L_1(I + DL_1)^{-1}, \\ \widetilde{K}_2 &= [I - L_1(I + DL_1)^{-1}D]K_2 - L_1(I + DL_1)^{-1}D_0, \end{aligned} \tag{4.76}$$

for any L_1 such that $I+DL_1$ is nonsingular, and the following result is immediate to be presented.

Theorem 4.21 *Suppose that the differential LRP of the form described by (2.14)-(2.15) is subject to a control law of the form (4.71) and that (4.75) holds. Then the resulting closed loop process is stable along the pass if there exist matrices $Y > 0$, $Z > 0$, $X > 0$ and N such that the following LMI holds*

$$\begin{aligned} \begin{bmatrix} YA^T + AY + C^TN^TB^T + BNC & B_0Z + BM & YC^T + C^TN^TD^T \\ ZB_0^T + M^TB^T & -Z & ZD_0^T + M^TD^T \\ CY + DNC & D_0Z + DM & -Z \end{bmatrix} &< 0, \\ XC &= CY. \end{aligned} \tag{4.77}$$

If this condition holds, then the control law matrices L_1 and K_2 are given by

$$L_1 = NX^{-1}, \;\; K_2 = MZ^{-1} \tag{4.78}$$

and it is required that $I + DL_1$ is nonsingular. To compute the output controllers it is necessary to apply (4.76)

Proof. From (4.78), it is clear that $N = L_1X$ and substitution into the LMI of (4.77) with $XC = CY$ applied, gives

$$\begin{bmatrix} Y(A^T + C^TL_1^TB^T) + (A + BL_1C)Y & B_0Z + BM & Y(C^T + C^TL_1^TD^T) \\ ZB_0^T + M^TB^T & -Z & ZD_0^T + M^TD^T \\ (C + DL_1C)Y & D_0Z + DM & -Z \end{bmatrix} < 0.$$

Finally, set $L_1C = K_1$ to obtain the following stabilization condition (i.e. Theorem 4.8 applied for the closed loop process)

$$\begin{bmatrix} Y(A^T + K_1^TB^T) + (A + BK_1)Y & (B_0 + BK_2)Z & Y(C^T + K_1^TD^T) \\ Z(B_0^T + K_2^TB^T) & -Z & Z(D_0^T + K_2^TD^T) \\ (C + DL_1C)Y & (D_0 + DK_2)Z & -Z \end{bmatrix} < 0,$$

which completes the proof. □

Example 4.13 *To present the applicability of Theorem 4.21 consider again the model of unstable differential LRP of (2.14)-(2.15), given already in Example 4.5. The application of LMI of (4.77) provides the following matrices*

$$Y = \begin{bmatrix} 3655.0895 & 1147.6759 & -2149.7151 & 2818.2736 \\ 1147.6759 & 12184.4771 & -2728.2939 & 6961.7644 \\ -2149.7151 & -2728.2939 & 3941.0220 & -3581.9408 \\ 2818.2736 & 6961.7644 & -3581.9408 & 8375.9234 \end{bmatrix},$$

$$X = \begin{bmatrix} 4276.9989 & 6478.9114 \\ 6478.9114 & 16645.8298 \end{bmatrix},\ Z = \begin{bmatrix} 9075.6366 & -1597.9599 \\ -1597.9599 & 14232.5180 \end{bmatrix},$$

$$N = \begin{bmatrix} -1986.9541 & -15060.1217 \\ 1457.1815 & 10696.0362 \\ 772.1232 & 1991.0073 \end{bmatrix},\ M = \begin{bmatrix} 366.6512 & -3468.5115 \\ -1386.4838 & 2340.5496 \\ -904.3451 & -242.8691 \end{bmatrix}.$$

Hence

$$L_1 = \begin{bmatrix} 2.2075 & -1.7639 \\ -1.5416 & 1.2426 \\ -0.0016 & 0.1202 \end{bmatrix}$$

and the state/output controllers become

$$K_1 = \begin{bmatrix} -1.1038 & 0.1754 & -1.1471 & 0.8376 \\ 0.7708 & -0.1289 & 0.8054 & -0.5914 \\ 0.0008 & -0.0715 & 0.0484 & -0.0720 \end{bmatrix},\ K_2 = \begin{bmatrix} -0.0026 & -0.2440 \\ -0.1263 & 0.1503 \\ -0.1047 & -0.0288 \end{bmatrix}.$$

Finally, the output controllers computed according to (4.76) become

$$\widetilde{K}_1 = \begin{bmatrix} 67.1230 & -11.4202 \\ -47.9335 & 8.1772 \\ -11.7664 & 2.2440 \end{bmatrix},\ \widetilde{K}_2 = \begin{bmatrix} 3.8255 & 1.5022 \\ -2.8554 & -1.0938 \\ -0.7245 & -0.3024 \end{bmatrix}.$$

Remark 4.11 *Note that for the differential LRPs, it is possible to use the extended set of the output controllers as well. It can be done in the similar manner to that presented for the discrete LRPs in Sections 4.9.1-4.9.2. This could again lead to the conservativeness reduction in exactly the same way as it has a place in the discrete case.*

The output controller design with application of parallel computing

Note that the application of the LMI conditions aforementioned may cause the unintended increase of the number of the decision variables. The second aspect of that control scheme regards the use of extended output control approaches and what follows, the growth of the total size of the LMI. Hence it appears to be quite natural, to employ the computational power, which is governed by the use of the parallel computing techniques. However, it is to note that there arises a serious obstacle in using the parallel SDP software (SDPARA), since in the considered output based synthesis the equality constraints appear. It is straightforward to see that the definition of SDP problem of (3.10), given in Section 3.5, does not involve the equality constraints. Hence the elimination of the LMI with the equality constraints into the one without them is required (as presented in Section 3.6), before performing the construction of the LMI into the valid SDP form (as presented in Section 3.5).

The procedure presented in Section 3.6 can be successfully applied to the considered here output controller design. To perform the elimination of the equality constraints, first note that $XC = CY$ (in all considered cases) has to be rewritten as $XC - CY = 0$. Then note that matrices X and Y are constructed of the decision variables over which the LMI is solved. However, it is necessary to remain that there are also additional decision variables, i.e. those of which matrices Z and N are constructed. What is important here that vector b in this case equals 0, hence the appropriate mapping can be presented as follows

$$F(y) = F(H^\sharp(x)) < 0.$$

To solve such a problem using the SDP solver, when the equality constraints have been eliminated, from the resulting LMI of $F(y)$ the valid form of SDP (according to procedure presented in Section 3.5) has to be performed. Then after computing the vector y, it has to be re-mapped back into x, matrices of the original LMI (here X, Y, Z, N) have to be constructed and, finally, the controller matrices have to be computed.

The appropriate MATLAB function regarding the extended control scheme considered in Section 4.9.2, which performs all required steps of equality constraints elimination and saves the resulted problem into the file as a valid SDP problem, has been attached in Appendix B.3.

4.10 Model Matching Based Controller Design

The controller design procedures outlined in Sections 4.2 and 4.4 guarantee the closed loop stability (asymptotic or/and along the pass) but not resulting closed loop dynamics. Here, new results are given, which address the currently open question of how to design a control laws for discrete LRPs for both closed loop stability (asymptotic or along the pass) and obtaining the assumed reference model.

4.10.1 1D model matching (asymptotic stability)

Model following control is a long standing technique in 1D systems theory and there has also been some work on this problem for 2D discrete linear systems described by the Roesser and Fornasini state-space models [24]. In the remainder of this section, some new results are presented, which provide a possible starting point for the development of a mature model following control theory for discrete LRPs.

Note that since in this section, asymptotic stability is considered, the 1D equivalent model of LRP described by (2.24)-(2.25) (provided for the basic discrete LRP of (2.9)-(2.10) or generalized LRP of (2.19)-(2.20)) is now investigated. First, note that in (2.25), the system matrix Φ describes the contribution of the previous pass profile to the current one. Also, under the action of the control law (4.3), this matrix is 'mapped' as follows

$$\Phi \rightarrow \Phi + \Delta K.$$

Due to the fact that the considered model on average is large-dimensioned, the known 1D techniques (see e.g. [45, 143]) of poles placement can fail or in general be disturbed. Instead, the following solution can be proposed.

Suppose first that the additional goal of the controller design over ensuring asymptotic stability is now to provide such a controller to assign the closed loop system matrix to become a'priori set matrix, say $\widehat{\Phi}$, which guarantees the required closed loop process dynamics. This matrix is selected to give a state-space model whose behavior is such that, it satisfies at least some of the overall process dynamical specifications.

Hence the following easily proved result is relevant.

Theorem 4.22 *Suppose that a discrete LRP (2.9)-(2.10) or generalized LRP (2.19)-(2.20))) is given in the 1D equivalent model of (2.24)-(2.25) and is subject to a control law defined in terms of the 1D equivalent model by (4.3). Then the resulting closed loop process is asymptotically stable with the required matrix $\widehat{\Phi}$ if there exist matrices $P > 0$, G and N such that the following LMI holds*

$$\begin{bmatrix} -P & (\Phi - \widehat{\Phi})G + \Delta N \\ G^T(\Phi - \widehat{\Phi})^T + N^T\Delta^T & P - G - G^T \end{bmatrix} < 0, \tag{4.79}$$

where Φ and Δ are of the structures defined in (2.26) or (2.27). Also the control law matrix K here is computed using (4.7) in such a way that

$$\Phi - \widehat{\Phi} + \Delta K = 0. \tag{4.80}$$

Proof. First note again that, if the LMI (4.79) holds then the control law matrix K of (4.3) is given by (4.7). Also, it is a standard fact that it is possible to obtain from the LMI solver K such that (4.80) holds. In such a case, the closed loop system matrix becomes

$$\widetilde{\Phi} := \Phi + \Delta K = \widehat{\Phi},$$

which completes the proof. □

It is essential to note here that it is impossible to obtain an arbitrarily specified model matrix $\widehat{\Phi}$ starting from an arbitrary specified Φ. However, conditions under which (4.80) has a solution can be characterized easily starting from, for example, Cramer's rule for linear vector equations and the matrix Kronecker matrix product.

In terms of 1D system this case can be treated as one of wide range of poles location technique.

Remark 4.12 *The importance of the above theorem comes from the fact that when the LMI techniques are used to synthesis, there seldom happens that the computed controller drives the closed loop system to zero, i.e. $\Phi + \Delta K = 0$. This, in general, is referred to be negative and in the classical 1D theory is so-called the deadbeat control. In terms of physical systems this situation denotes application of the control of the high magnitude.*

The applicability of the model matching approach presented here is provided by the following Example. This approach has been also applied in Example 5.3.

Example 4.14 *Consider the following model of (2.19)-(2.20), with $\alpha = 8$*

$$A = \begin{bmatrix} 0.2 & 0 & 0.5 \\ 0 & 0.5 & 0 \\ 0.5 & 0 & -0.5 \end{bmatrix},\ B = \begin{bmatrix} 0.2 & 0.7 \\ 0.5 & 1.0 \\ 0.5 & 0.7 \end{bmatrix},\ C = \begin{bmatrix} 0.1 & 0 & 0.6 \end{bmatrix},\ D = \begin{bmatrix} 0.7 & 0.9 \end{bmatrix},$$

$$\mathbf{B} = \left[\begin{array}{c|c|c|c} B_0 & B_1 & \dots & B_7 \end{array}\right] = \left[\begin{array}{c|c|c|c|c|c|c|c} 0 & 0.4 & -0.5 & 0 & 0 & 0 & 0.3 & 0.2 \\ -1.5 & -0.2 & -0.1 & -0.3 & 0 & 0 & 0 & 0.1 \\ 1.9 & 0.7 & 0.4 & -0.2 & 0.1 & 0.1 & 0.3 & -0.2 \end{array}\right],$$

$$\mathbf{D} = \left[\begin{array}{c|c|c|c} D_0 & D_1 & \dots & D_7 \end{array}\right] = \left[\begin{array}{c|c|c|c|c|c|c|c} 2.3 & 0 & 0.9 & 0.2 & 0.2 & -0.5 & 0.2 & 0.3 \end{array}\right].$$

The required system matrix has been set to be $\widetilde{\Phi} = diag(0.8,\ 0.7,\ 0.6,\ 0.5,\ 0.4,\ 0.3,\ 0.2,\ 0.1)$. *The application of Theorem 4.22 provides the following controller*

$$K = \begin{bmatrix} -0.7672 & 0.0091 & -0.5055 & -0.0691 & -0.1198 & 0.2574 & -0.1054 & -0.1380 \\ -1.0700 & -0.0071 & -0.6068 & -0.1685 & -0.1290 & 0.3553 & -0.1402 & -0.2260 \\ -1.2692 & 0.1512 & -0.2949 & -0.0226 & -0.0578 & 0.0796 & -0.1376 & -0.0273 \\ -1.9798 & 0.1497 & -0.4716 & 0.0450 & -0.1311 & 0.1420 & -0.2347 & -0.0288 \\ -0.8009 & -0.2686 & 0.0622 & -0.0643 & -0.0984 & 0.1915 & -0.1423 & -0.1376 \\ -1.1361 & -0.3842 & 0.0565 & -0.1064 & -0.0938 & 0.2602 & -0.1910 & -0.2135 \\ -1.1490 & -0.1487 & -0.4169 & 0.2821 & -0.0700 & 0.0579 & -0.1458 & -0.0206 \\ -1.7799 & -0.2311 & -0.6268 & 0.3712 & -0.1093 & 0.1287 & -0.2424 & -0.0317 \\ -0.7554 & -0.1994 & -0.1639 & -0.2024 & 0.1384 & 0.1809 & -0.1450 & -0.1434 \\ -1.1166 & -0.2879 & -0.2447 & -0.2779 & 0.1649 & 0.2645 & -0.1929 & -0.2088 \\ -1.1674 & -0.1328 & -0.3610 & 0.0903 & -0.1779 & 0.2300 & -0.1564 & -0.0163 \\ -1.7544 & -0.2136 & -0.5318 & 0.1108 & -0.2498 & 0.3240 & -0.2340 & -0.0345 \\ -0.7544 & -0.2041 & -0.1456 & -0.1624 & -0.0191 & 0.1081 & -0.0290 & -0.1574 \\ -1.1149 & -0.2795 & -0.2312 & -0.1977 & -0.0408 & 0.1530 & -0.0627 & -0.1988 \\ -1.2917 & -0.1528 & -0.3921 & 0.0993 & -0.1448 & 0.1240 & -0.2270 & 0.0312 \\ -1.6607 & -0.1964 & -0.5041 & 0.1277 & -0.1862 & 0.1594 & -0.2919 & 0.0401 \end{bmatrix}.$$

4.10.2 2D model matching (stability along the pass)

Similarly as it has a place for 1D equivalent model and the concept of asymptotic stability, the model matching technique can be defined as well for the concept of stability along the pass. It is even more meaningful here, since there are no results regarding the 2D models poles placement available.

To provide the 2D model matching, note that the state-space quadruple $\{A, B_0, C, D_0\}$ describes the contribution of the previous pass profile to the current one. Also, under the action of the control law (4.11) this quadruple is 'mapped' as follows

$$\begin{bmatrix} A & B_0 \\ C & D_0 \end{bmatrix} \rightarrow \left[\begin{array}{c|c} A + BK_1 & B_0 + BK_2 \\ \hline C + DK_1 & D_0 + DK_2 \end{array}\right].$$

Suppose also that the synthesis goal here is to assign the closed loop matrices here to $\{\mathcal{A}, \mathcal{B}_0, \mathcal{C}, \mathcal{D}_0\}$, where these matrices are selected to give a state-space model whose behavior the controlled process is required to follow (in terms of the contribution of the previous pass profile to the current one). Then the following result is relevant.

Theorem 4.23 *Suppose that a discrete LRP of (2.9)-(2.10) is subjected to a control law of the form (4.11). Then the resulting closed loop process is stable along the pass and reaches the required form* $\{\mathcal{A}, \mathcal{B}_0, \mathcal{C}, \mathcal{D}_0\}$ *if there exist matrices* $Y > 0$, $Z > 0$ *and* N *of the appropriate* *dimensions such that the following LMI holds*

$$\begin{bmatrix} Z-Y & 0 & Y\widetilde{A}_1^T + N^T\hat{B}_1^T \\ 0 & -Z & Y\widetilde{A}_2^T + N^T\hat{B}_2^T \\ \widetilde{A}_1 Y + \hat{B}_1 N & \widetilde{A}_2 Y + \hat{B}_2 N & -Y \end{bmatrix} < 0, \tag{4.81}$$

$$\text{where} \quad \widetilde{A}_1 = \begin{bmatrix} A-\mathcal{A} & B_0-\mathcal{B}_0 \\ 0 & 0 \end{bmatrix}, \widetilde{A}_2 = \begin{bmatrix} 0 & 0 \\ C-\mathcal{C} & D_0-\mathcal{D}_0 \end{bmatrix}$$

and the other matrices are the same as in the previous cases, where the goal of controller design towards stability along the pass was considered. If condition (4.81) holds, then the required controllers K_1 and K_2 are computed using (4.14).

Proof. First note again that, if the LMI (4.12) holds, then the control law matrix $K = [K_1\, K_2]$ is given by (4.14). Also, it is a standard fact that it is possible to obtain from the LMI solver matrix K such that

$$\left[\begin{array}{c|c} A-\mathcal{A} & B_0-\mathcal{B}_0 \\ \hline C-\mathcal{C} & D_0-\mathcal{D}_0 \end{array}\right] + \left[\begin{array}{c} B \\ \hline D \end{array}\right] [K_1\, K_2] = 0 \tag{4.82}$$

holds. In which case, the closed loop system matrices are such that

$$\left[\begin{array}{c|c} \tilde{A} & \tilde{B}_0 \\ \hline \tilde{C} & \tilde{D}_0 \end{array}\right] := \left[\begin{array}{c|c} A+BK_1 & B_0+BK_2 \\ \hline C+DK_1 & D_0+DK_2 \end{array}\right] = \left[\begin{array}{c|c} \mathcal{A} & \mathcal{B}_0 \\ \hline \mathcal{C} & \mathcal{D}_0 \end{array}\right],$$

which completes the proof. □

It is essential to underline here that it is impossible to obtain an arbitrarily specified set $\{\mathcal{A}, \mathcal{B}_0, \mathcal{C}, \mathcal{D}_0\}$ starting from a given set $\{A, B_0, C, D_0\}$. However, conditions under which (4.82) has a solution can be characterized using, for example, Cramer's rule for linear vector equations and the matrix Kronecker matrix product.

The result of Theorem 4.23 is appreciated especially in the case of synthesis of 2D systems, where there are no explicit connections between the properties of the system and the poles of the transfer function. As it was shown e.g. in [50], poles of the polynomial of two (or more) variables become the continuous functions located on the complex plane. Hence for 2D systems, in most cases it is impossible to analyze the problem of poles location, using 1D theory based techniques.

For this approach there also arises the case considered before for asymptotic stability regarding the issues of the deadbeat control.

Example 4.15 *To present how Theorem 4.23 works consider the model of physical process of metal rolling presented in Section 2.5.1. The process is modeled as discrete LRP with the state-space representation of (2.35). Note that the basic stabilization of Theorem 4.9 provides the following controllers*

$$K_1 = \begin{bmatrix} 86.667 & -37.372 & -66.667 & 33.333 \end{bmatrix}, \; K_2 = \begin{bmatrix} 35.333 \end{bmatrix},$$

which causes the closed loop model becomes so-called "zero-model", i.e. (in this case)

$$\left[\begin{array}{c|c} A+BK_1 & B_0+BK_2 \\ \hline C+DK_1 & D_0+DK_2 \end{array}\right] = \left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 0 \\ 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 1.0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

Then in this case the application of the 2D model matching procedure of Theorem 4.23 is necessary with chosen closed loop required plant matrix defined as

$$\left[\begin{array}{c|c} \mathcal{A} & \mathcal{B}_0 \\ \hline \mathcal{C} & \mathcal{D}_0 \end{array}\right] = \left[\begin{array}{cccc|c} 0.1 & -0.1 & 0.1 & 0.1 & 0.7 \\ 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 1.0 & 0 & 0 \\ \hline 0.1 & -0.1 & 0.1 & 0.1 & 0.7 \end{array}\right]. \tag{4.83}$$

The LMI (4.81) provides the following controllers

$$K_1 = \left[\begin{array}{cccc} 82.13 & -32.84 & -71.2 & 28.8 \end{array}\right],\ K_2 = \left[\begin{array}{c} 3.6 \end{array}\right]$$

and it can be easily checked that the closed loop model plant matrix with the above controllers applied becomes (4.83) as it was required.

Chapter 5

Control for performance

In the practical control schemes the goal of ensuring the stability in the closed loop is too less. Now, when the LMI conditions for determining the controllers (towards asymptotic stability and/or stability along the pass) have been presented in the previous chapter, it is natural to extend the synthesis of LRP to the performance requirements under the appropriate control.

To formalize the concept of performance used here, the goals of the considered in this chapter control schemes are defined as follows:

- the stability (asymptotic or along the pass) in the closed loop system configuration,
- after the sufficient large number of passes the process is driven to the required reference signal $y_{ref}(p)$, $0 \leq p \leq \alpha - 1$ ($y_{ref}(t)$, $0 \leq t < \alpha$),
- rejection of the disturbances that influence the controlled LRP.

In this chapter, the disturbed state-space models of LRPs are considered. Hence discrete LRP of (2.9)-(2.10) now has the following form over $0 \leq p \leq \alpha - 1$

$$x_{k+1}(p+1) = Ax_{k+1}(p) + B_0 y_k(p) + Bu_{k+1}(p) + Ew(p), \tag{5.1}$$

$$y_{k+1}(p) = Cx_{k+1}(p) + D_0 y_k(p) + Du_{k+1}(p) + Fw(p) \tag{5.2}$$

and differential LRP of (2.14)-(2.15) has the following form over $0 \leq t < \alpha$ respectively

$$\dot{x}_{k+1}(t) = Ax_{k+1}(t) + B_0 y_k(t) + Bu_{k+1}(t) + Ew(t), \tag{5.3}$$

$$y_{k+1}(t) = Cx_{k+1}(t) + D_0 y_k(t) + Du_{k+1}(t) + Fw(t). \tag{5.4}$$

There should be underlined that in this case, it is assumed that disturbances do not change in the from pass to pass (k) direction, i.e. $w_{k_1}(p) = w_{k_2}(p),\ 0 \leq p \leq \alpha - 1$ ($w_{k_1}(t) = w_{k_2}(t),\ 0 \leq t < \alpha$) for any two pass numbers $k_1,\ k_2 \in Z$. Nevertheless, the disturbances can be dynamic in the along the pass direction p (or t). Due to that assumption the disturbance vector is denoted as $w(p)$ ($w(t)$) (without explicit number of pass given).

It is also important to underline that the assumed performance objectives are by no means exhaustive and what is being undertaken here is an examination of the feasibility of designing one possible control law structure.

Results presented in this chapter are the basis for the following publications (already published or being at the process of publication) – see the references: [25, 144] – Sections 5.1 and 5.2; [25, 133, 145] – Section 5.3 and [146, 147, 138, 148] – Section 5.4 and are original author's results.

5.1 Direct control scheme

As the first approach to obtain the reference signal $y_{ref}(p),\ 0 \leq p \leq \alpha - 1$ for LRPs of (5.1)-(5.2) in the closed loop system the following control law is considered

$$u_{k+1}(p) = K_1 x_{k+1}(p) + K_2 y_k(p) + K_3 r_{k+1}(p), \tag{5.5}$$

where K_1 and K_2 are defined (and have the meaning) as in (4.11), i.e. are about to ensure the stability along the pass and additionally factor $K_3 r_{k+1}(p),\ 0 \leq p \leq \alpha - 1$ is added to drive the closed loop system to the required reference signal. Such an approach is common with 1D linear systems (see e.g. [37]). Here, only first two of the goals presented in the introduction of that chapter are assumed to be satisfied, i.e. it is assumed that there are no disturbances influencing the considered process (i.e. $w = 0$).

The sequence $r_{k+1}(p)$ is an $m \times 1$ column vector representing desired the behavior on pass $k+1,\ k \geq 0$, and K_3 is an $r \times m$ controller matrix to be selected. It is assumed that $\lim_{k\to\infty} K_3 r_{k+1}(p) = y_{ref}(p),\ 0 \leq p \leq \alpha - 1$. Since the closed loop system is stable along the pass, output of the system should reach the defined additional factor $K_3 r_{k+1}(p)$ (at last $y_{ref}(p)$). The application of (5.5) results the following closed loop state-space model

$$\begin{aligned} x_{k+1}(p+1) &= (A + BK_1)x_{k+1}(p) + (B_0 + BK_2)y_k(p) + BK_3 r_{k+1}(p), & (5.6)\\ y_{k+1}(p) &= (C + DK_1)x_{k+1}(p) + (D_0 + DK_2)y_k(p) + DK_3 r_{k+1}(p). & (5.7) \end{aligned}$$

Obvious questions which now arise are:

- what is a suitable choice for $r_{k+1}(p)$?
- how the appropriate control law can be designed to give stability along the pass plus performance?

The answer for the first question from the above has been already partially given, i.e. it has been stated that the following should satisfy $\lim_{k\to\infty} K_3 r_{k+1}(p) = y_{ref}(p),\ 0 \leq p \leq \alpha - 1$. This condition allows to change the value of the signal $r_{k+1}(p)$ when k increases but in practise, it is assumed that the whole sequence $r_{k+1}(p)$ is swapped with $y_{ref}(p)$ independently of current pass number k. There still is the open question about the choosing the appropriate reference signal $y_{ref}(p)$ and this should be chosen as a value fulfilling the physical requirements of the considered system.

The answer for the second question regarding the design of the controllers applicable in (5.5) has been given in the previous chapter when the controller design procedure has been presented (Section 4.4) or in its "improved" version e.g. the controller design to required stability margins (Section 4.5) or the model matching procedure (Section 4.10.2). The case of choosing the controller K_3 is still open and there are no ready solutions applicable in every case.

To illustrate, what can be achieved here, the emphasis is put on the single-input single-output case and use the metal rolling problem data, aforementioned in Section 2.5.1. In this application, an appropriate choice for the current pass reference signal is $r_{k+1}(p) = -1,\ 0 \leq p \leq \alpha - 1,\ k \geq 0$, i.e the objective is to reduce the material thickness by one unit, which is modeled by a downward unit step applied at $p = 0$ on each pass.

Remark 5.1 *Note that since the process is linear, any target reduction by a constant amount can be studied by simple scaling of the output pass profiles to a unit step demand.*

One possible way of designing the control law is to note that K_3 does not influence stability along the pass. Hence it is possible to execute the LMI design of Section 4.10.2 to obtain control law matrices K_1 and K_2, which ensure closed loop stability along the pass and then attempt to select a suitable K_3 to meet the performance requirements by 'tuning' the response of the resulting closed loop process model.

Example 5.1 *In the case of the given numerical data (metal rolling process defined in Section 2.5.1), it is easily checked that this model is unstable along the pass and the stabilization procedure of Theorem 4.23 (Section 4.10.2) provides the following control law matrices* K_1 *and* K_2

$$K_1 = \begin{bmatrix} 82.13 & -32.84 & -71.2 & 28.8 \end{bmatrix}, \ K_2 = \begin{bmatrix} 3.6 \end{bmatrix}$$

and in the resulting stable along the pass closed loop process

$$\left[\begin{array}{c|c} \mathcal{A} & \mathcal{B}_0 \\ \hline \mathcal{C} & \mathcal{D}_0 \end{array}\right] = \left[\begin{array}{cccc|c} 0.1 & -0.1 & 0.1 & 0.1 & 0.7 \\ 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 1.0 & 0 & 0 \\ \hline 0.1 & -0.1 & 0.1 & 0.1 & 0.7 \end{array}\right].$$

Figures 5.1 a) and b) show the sequence of pass profiles and the tracking error on each pass for the case when $K_3 = -4.53$ *and with zero initial/boundary conditions assumed where this value was derived by repeated numerical experimentation with the objective of obtaining the smallest error between* $r_{k+1}(p)$ *and* $y_k(p)$ *anywhere in the domain of operation. This can be treated as an*

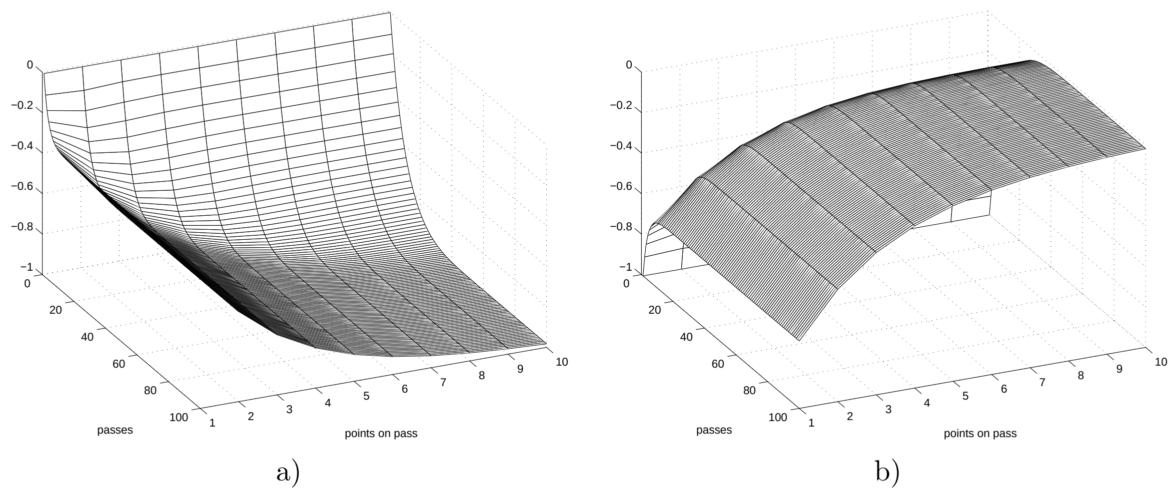


Figure 5.1. Pass profiles (a) and tracking error (b)

acceptable design, especially as it does not contain any oscillations in the transients along any pass (not a desirable feature in material rolling, however it is straightforward to see that for the first 50 passes the tracking error is meaningful and what is more it still is relatively large at the beginnings of all passes.

When for the same process the nonzero initial state vectors have been assumed (here chosen arbitrary as $d_{k+1} = [-0.71 \ \ 0.78 \ \ -0.7 \ \ 0.19]^T, \ \forall k$*) the dynamics of the process under control*

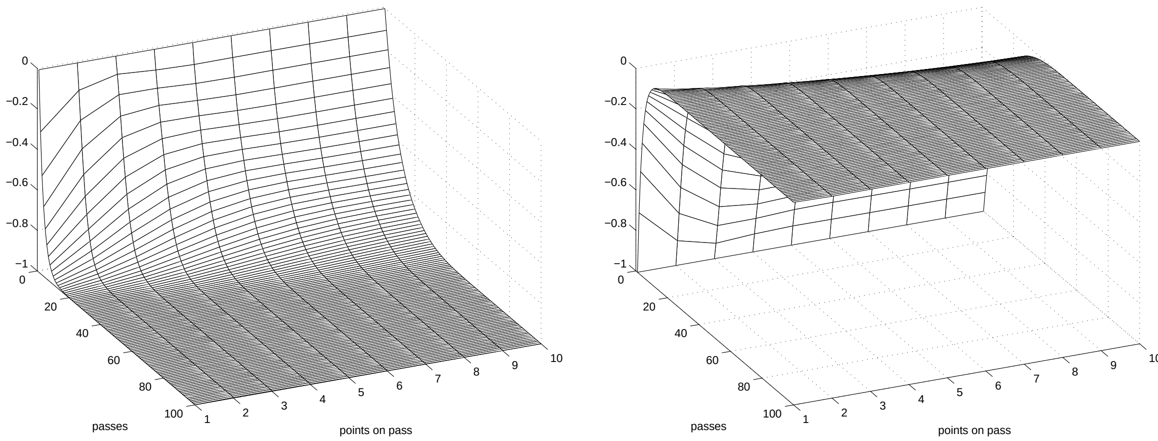


Figure 5.2. Pass profiles (a) and tracking error (b)

is more effective (the tracking error is significantly smaller, especially at the beginnings of the following passes) - see Figures 5.2 a) and b).

5.2 Indirect control scheme

The empirical nature of the previous approach, however, means that it is clearly not feasible in the general case. Next a systematic method for the task under consideration is developed and again is illustrated on the material rolling problem data. Again only first two goals presented in the introduction are to be satisfied here, i.e. there are no disturbances influencing the system.

This new approach is based on a (simple structure) re-formulation of the problem, starting from the fact that the control task here is to drive the process pass profiles to some prescribed reference signal $y_{ref}(p)$, which in the material rolling case is a constant positive thickness after the rolling operation is complete, i.e. $y_{ref}(p),\ 0 \leq p \leq \alpha - 1$ and in the general case can be any profile that can be thought. It is an immediate consequence (see Section 4.3) of the stability theory that if asymptotic stability holds then the pass profile sequence converges to a steady, or so-called limit, profile described for discrete linear repetitive processes by a 1D linear systems state-space model. Here it is necessary to specify this limit profile as $y_{ref}(p)$.

To solve this last problem introduce a new, modified output vector variable termed the incremental (or residual) pass profile vector as

$$\hat{y}_k(p) := y_k(p) - y_{ref}(p). \tag{5.8}$$

Then it is clear that the design requirement (stability along the pass) here requires that

$$\hat{y}_k(p) \to 0,\ 0 \leq p \leq \alpha - 1,\ k \to \infty. \tag{5.9}$$

Now replace the process state-space model (5.1)-(5.2) (without disturbances as mentioned) by the following one obtained from it by substitution using (5.8)

$$\begin{aligned} x_{k+1}(p+1) &= Ax_{k+1}(p) + Bu_{k+1}(p) + B_0\hat{y}_k(p), \\ \hat{y}_{k+1}(p) &= Cx_{k+1}(p) + Du_{k+1}(p) + D_0\hat{y}_k(p). \end{aligned} \tag{5.10}$$

and apply to it the following control law (which is clearly of the form (4.11), i.e. current pass state feedback augmented in this case by feedforward of the difference between $y_k(p)$ and $y_{ref}(p)$)

$$\begin{aligned} u_{k+1}(p) &= K_1 x_{k+1}(p) + K_2 \hat{y}_k(p) \\ &= K_1 x_{k+1}(p) + K_2(y_k(p) - y_{ref}(p)). \end{aligned}$$

Also choose this control law to transform the process of (5.10) into the form of those for which any of the controller design result e.g. Theorem 4.23 holds. Then it follows immediately that this resulting closed loop model must be stable along the pass and also (5.9) holds. Moreover,

$$x_k(p) \to 0,\ 0 \leq p \leq \alpha - 1, k \to \infty, \tag{5.11}$$

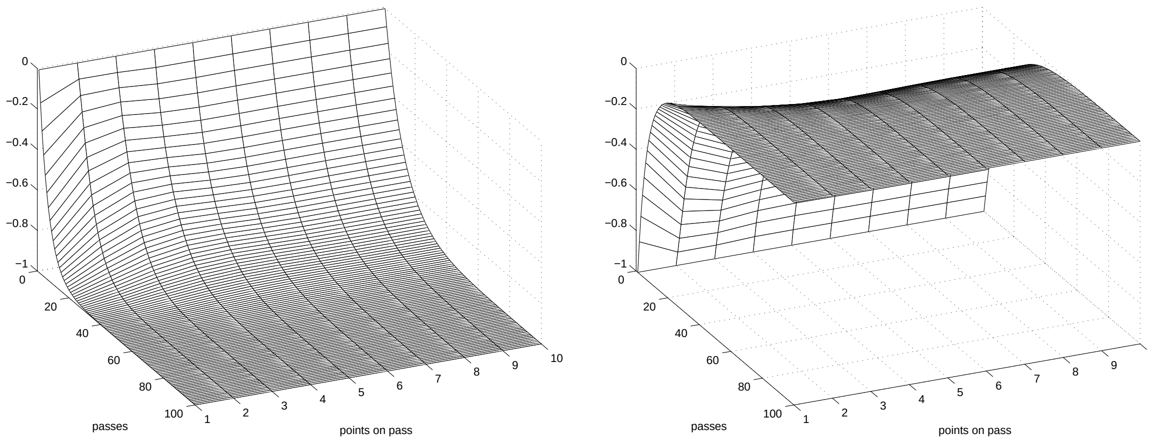
which is natural, and most frequently obtained result, when using the LMI based approach to controller design. Also, no oscillations can occur in the resulting pass profiles (which is clearly a required feature in the specific material rolling example considered here) since the model matching approach has been applied.

Given this designed feedback law and converting back to the original pass profile vector $y_k(p)$ the following resulting closed loop state-space model is obtained

$$\begin{aligned} x_{k+1}(p+1) &= (A + BK_1)x_{k+1}(p) + (B_0 + BK_2)(y_k(p) - y_{ref}(p)), \\ y_{k+1}(p) &= (C + DK_1)x_{k+1}(p) + (D_0 + DK_2)y_k(p) + (I - (D_0 + DK_2))y_{ref}(p), \end{aligned}$$

which is stable along the pass and whose limit pass profile, due to (5.9) and (5.11), is clearly equal to $y_{ref}(p)$, i.e. the control design task has been exactly achieved.

Example 5.2 *To illustrate this approach, the material rolling process described in Section 2.5.1 is considered again. The model and controllers are the same as in Example 5.1. To execute the presented design, the following reference signal has been assumed* $y_{ref}(p) = -1,\ 0 \leq p \leq \alpha - 1,\ k \geq 0$. *This produces the simulation results of Figures 5.3 a) and b) for the resulting*



a) b)

Figure 5.3. Pass profiles (a) and tracking error (b)

closed loop process in the case of zero boundary conditions and these confirm that the design objective has indeed been achieved.

What is worth to mention here is that the above approach is more robust to changes of the boundary conditions and even in the "worst" case (zero initial state vectors) works better in comparison with result from Section 5.1 (refer to Example 5.1).

Remark 5.2 *It is also possible to present the above control schemes for the differential LRP of (5.3)-(5.4). It is the straightforward operation and its only difference comes from the controller design conditions to be applied. Hence due the relatively poor performance (in comparison with results presented in the sequel of this chapter) is skipped here.*

5.3 Feedforward/Feedback control of LRPs

In this section, the basic design task is the development of the control laws which give asymptotic stability or stability along the pass and completely reject the effects of the disturbance is considered. What is more, in this case, it is assumed that the disturbance sequence is known.

5.3.1 Discrete LRPs – asymptotic stability – 1D equivalent model

The particular aim of the design is to achieve asymptotic stability and complete decoupling of the effects of the disturbance term $w(p)$ on the closed loop performance and simultaneously drive the process to the required reference signal V_{ref}, i.e. limit pass profile [17, 149] of the process defined in terms of the 1D equivalent model by $V_\infty := \underset{l\to\infty}{V(l)}$ has to be equal to the pre-specified vector V_{ref}.

To begin analysis, define V_{ref} as

$$V_{ref} = \begin{bmatrix} y_{ref}(0)^T & y_{ref}(1)^T & \dots & y_{ref}(\alpha-1)^T \end{bmatrix}^T$$

and next rewrite the 1D equivalent model equation (2.25) as

$$V(l+1) = \Phi V(l) + \Delta U(l) + \Theta_0 d_l + \Omega_y W. \tag{5.12}$$

Then the control task here is to ensure asymptotic stability closed loop and also

$$V(l) \to V_{ref},\ l \to \infty.$$

If asymptotic stability holds, i.e. $r(\Phi) < 1$, then

$$V(l+1) - V(l) \to 0,\ l \to \infty,$$

which is equivalent to

$$V(l+1) = V(l) \equiv V_s,\ l \to \infty, \tag{5.13}$$

where V_s denotes the limit profile and obviously, it is required that $V_s = V_{ref}$). Now apply (5.13) to (5.12) and, with U_s denoting the control input vector applied on the limit profile, (5.12) converges to

$$V_{ref} = \Phi V_{ref} + \Delta U_s + \Theta_0 d_s + \Omega_y W. \tag{5.14}$$

where d_s denotes the initial state vector on the limit profile.

Subtracting (5.14) from (2.25) now yields

$$V(l+1) - V_{ref} = \Phi\big(V(l) - V_{ref}\big) + \Delta\big(U(l) - U_s\big) + \Theta_0(d_l - d_s), \tag{5.15}$$

and define the incremental (residuals) supervectors as

$$\begin{aligned}\hat{V}(l+1) &= V(l+1) - V_{ref},\\ \hat{U}(l+1) &= U(l+1) - U_s,\\ \hat{d}_l &= d_l - d_s.\end{aligned}$$

Hence (5.15) can be written in the following form

$$\hat{V}(l+1) = \Phi\hat{V}(l) + \Delta\hat{U}_l + \Theta_0\hat{d}_l. \tag{5.16}$$

The key point now is that the disturbance $w(p)$ has been decoupled from the dynamics of (5.16). Suppose, however that the dynamics of the resulting limit profile do not meet the specifications (for example, in the single-input single-output (SISO) case the dynamics are not 'sufficiently well' damped) then one option is to choose $\hat{U}(l)$ as a feedback control law of the form

$$\hat{U}(l) = K\hat{V}(l) \Rightarrow U(l) - U_s = K\Big(V(l) - V_{ref}\Big),$$

where K is designed as in Section 4.2 but using the incremental model (5.16).

It is now immediate that the control law which drives the original process to the demanded V_{ref} and simultaneously decouples the influence of the disturbance is

$$U(l) = KV(l) - KV_{ref} + U_s. \tag{5.17}$$

which is a combination of feedback action and feedforward action (from pass-to-pass). The only remaining task to enable implementation of (5.17) is the need to explicitly compute U_s. This can be undertaken by rewriting (5.14) as

$$\Delta U_s = (I - \Phi)V_{ref} - \Omega_y W - \Theta_0 d_s \tag{5.18}$$

and it follows immediately that not all possible V_{ref} can be obtained by a valid control sequence. In particular, only those V_{ref} which satisfy

$$\text{rank}\,\big[\Delta \mid (I - \Phi)V_{ref} - \Omega_y W - \Theta_0 d_s\big] = \text{rank}\,\big[\Delta\big], \tag{5.19}$$

can be obtained. Due to the structure of Δ, it is immediate that if D (the matrix which describes how the current pass input vector couples to the current pass profile in (5.1)-(5.2)) is a full row rank matrix, then (5.18) is solvable and the solution is

$$U_s = \Delta^{\sharp}\Big((I - \Phi)V_{ref} - \Omega_y W - \Theta_0 d_s\Big), \tag{5.20}$$

where $(\cdot)^{\sharp}$ denotes the pseudo-inverse of $(\cdot)$.

Note that, if $m = r$, i.e. Δ is square and also nonsingular matrix, it is possible to replace the pseudo-inverse with the inverse and then (5.20) becomes

$$U_s = \Delta^{-1}\Big((I - \Phi)V_{ref} - \Omega_y W - \Theta_0 d_s\Big). \tag{5.21}$$

Note that if D is square and nonsingular then (5.21) is valid in this case. In the general case, however, it will often be the case that the dimension of the pass profile vector exceeds that of the input vector and hence neither D or Δ are full row rank matrices. As noted before, not all possible limit profile vectors can be achieved under this design. There is, however, a way of reducing the limitations this may impose. This is due to the fact that it may be possible to choose d_∞ to ensure that (5.19) holds.

Example 5.3 *The material rolling process given before is considered. Note that this process is asymptotically stable open loop since this property requires that* $r(D_0) < 1$ *and here for positive* $M,\ \lambda_1,\ \lambda_2$

$$D_0 = \frac{T^2 + \frac{M}{\lambda_1}}{T^2 + \frac{M}{\lambda_1} + \frac{M}{\lambda_2}} < 1 \rightarrow r(D_0) < 1.$$

The matrices are assumed as follows

$$E = [1\ 0\ 0\ 0]^T, \quad F = 1.$$

Suppose, however that other performance specifications do not hold. Then it is assumed that the model matching procedure presented formerly in Section 4.10.1 is applied with the reference 2D model defined as in Example 4.15. Hence the 1D reference model is constructed and the appropriate controller is to be computed. The controller matrix K *used for the feedforward feedback control scheme of (5.17) computed according to Theorem 4.22 has the form*

$$K = \left[\begin{array}{cccc} K^1 & K^2 & \dots & K^{20} \end{array}\right] = \left[\begin{array}{ccccc} k_1 & 0 & 0 & \dots & 0 \\ k_2 & k_1 & 0 & \dots & 0 \\ k_3 & k_2 & k_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{20} & k_{19} & k_{18} & \dots & k_1 \end{array}\right],$$

where

$$K^1 = \left[\begin{array}{cccccccccccccc} 3.6 & -13.71 & 19.78 & -1.72 & -2.55 & -0.08 & 0.25 & 0.03 & -0.02 & -0.01 & 0.01 & 0 & \dots & 0 \end{array}\right]^T.$$

The required output (reference signal) is assumed to be $y_{ref}(p) = -1\ \forall p$ *and the boundary conditions are zero (*$x_{k+1}(p) = [0\,0\,0\,0]^T$, $k \geq 0$ *and* $y_0(p) = 0,\ 0 \leq p \leq \alpha - 1$*). Note that the negative downward step here is to agree with the physical fact that the thickness of the bar is reduced on each successive pass through the rolling mechanism. It is also assumed that the disturbance sequence* $w(p)$ *is as shown in Figure 5.4 a), i.e. a sine wave with amplitude* 0.5 *shifted by* -2 *with some random value from the range* $(0, 1)$ *added.*

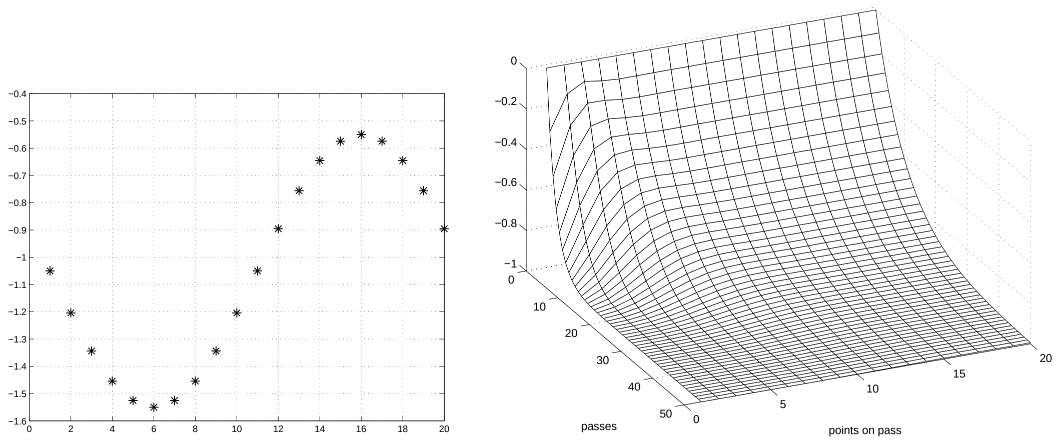
The pass length has been assumed to be $\alpha = 20$*, and the performance specification is that there are no oscillations in the dynamics along any pass (including the limit profile). The control vector here* U_s *is computed to be*

$$\begin{aligned} U_s \;=\; \Big[\; & 10 \quad -17 \quad -19.28 \quad -24.3 \quad -27.52 \quad -28.63 \quad -27.52 \quad -24.3 \quad -19.3 \quad -13 \\ & -5.96 \quad 1.04 \quad 7.36 \quad 12.38 \quad 15.6 \quad 16.71 \quad 15.6 \quad 12.38 \quad 7.36 \quad 1.04 \;\Big]^T. \end{aligned}$$

The output of the process with the control applied is shown in Figure 5.4 b) and the required performance specification is achieved.

5.3.2 Discrete LRPs – stability along the pass

Now the solution of the feedforward feedback control problem under stability along the pass, i.e. when asymptotic stability in the design objectives listed earlier in the previous section is replaced by the stronger requirement of stability along the pass.



a) b)

Figure 5.4. Disturbances (a) and pass profiles (b)

Suppose that stability along the pass holds. Then as $k \to \infty$ it is expected that for $0 \le p \le \alpha - 1$, $x_{k+1}(p) = x_k(p)$, $y_{k+1}(p) = y_{ref}(p)$ and hence

$$x_\infty(p+1) = Ax_\infty(p) + Bu_\infty(p) + B_0 y_{ref}(p) + Ew(p), \tag{5.22}$$

$$y_{ref}(p) = Cx_\infty(p) + Du_\infty(p) + D_0 y_{ref}(p) + Fw(p). \tag{5.23}$$

Now introduce

$$\begin{aligned} \hat{y}_k(p) &= y_k(p) - y_{ref}(p), \\ \hat{x}_k(p) &= x_k(p) - x_\infty(p), \\ \hat{u}_k(p) &= u_k(p) - u_\infty(p). \end{aligned}$$

Then subtract (5.22)-(5.23) from (5.1)-(5.2) to obtain

$$\hat{x}_{k+1}(p+1) = A\hat{x}_{k+1}(p) + B\hat{u}_{k+1}(p) + B_0\hat{y}_k(p), \tag{5.24}$$

$$\hat{y}_{k+1}(p) = C\hat{x}_{k+1}(p) + D\hat{u}_{k+1}(p) + D_0\hat{y}_k(p) \tag{5.25}$$

and in this so-called residual (incremental) model. Note that the influence of the disturbance vector has been completely decoupled.

Suppose now that a stabilizing control law is required. Then this can be achieved by writing

$$\hat{u}_{k+1}(p) = K_1\hat{x}_{k+1}(p) + K_2\hat{y}_k(p)$$

or, equivalently using the original variables,

$$u_{k+1}(p) = u_\infty(p) + K_1 x_{k+1}(p) - K_1 x_\infty(p) + K_2 y_k(p) - K_2 y_{ref}(p) \tag{5.26}$$

and the task now is to construct $x_\infty(p)$ and $u_\infty(p)$ when $w(p)$ and $y_{ref}(p)$ are known.

From the algebraic equation of (5.23) for $y_{ref}(p)$ it comes that

$$Du_\infty(p) = (I - D_0)y_{ref}(p) - Cx_\infty(p) - Fw(p).$$

Here, it is to note again (similarly as for the asymptotic stability case) that not all $y_{ref}(p)$ are available under this design. In particular, only those which ensure that

$$\text{rank}\,\big[D \mid (I - D_0)y_{ref}(p) - Cx_\infty(p) - Fw(p)\big] = \text{rank}\,\big[D\big] \tag{5.27}$$

are allowed. Suppose now that this condition holds (as is the case when D is full row rank). Then it is straightforward to see that

$$u_\infty(p) = D^\sharp\Big((I - D_0)y_{ref}(p) - Cx_\infty(p) - Fw(p)\Big). \tag{5.28}$$

Also if D is square and nonsingular, the pseudo inverse can be replaced by the matrix inverse and then

$$u_\infty(p) = D^{-1}\Big((I - D_0)y_{ref}(p) - Cx_\infty(p) - Fw(p)\Big). \tag{5.29}$$

Once $u_\infty(p)$ has been computed over $0 \le p \le \alpha - 1$ then

$$x_\infty(p+1) = (A - BD^{-1}C)x_\infty(p) + (B_0 + BD^{-1}(I - D_0))y_{ref}(p) + (E - BD^{-1}F)w(p) \tag{5.30}$$

(which is simply a 1D discrete linear systems state equation which can be computed given d_∞ — the strong limit of the known pass state initial vector sequence) and the overall control law is of the form (5.26) where K_1 and K_2 are the control law matrices computed by the LMI method to give stability along the pass closed loop.

5.3.3 Differential LRPs – stability along the pass

It is also possible to present the feedforward-feedback control scheme for the differential case of LRP. However, it is to note that there arises the serious problem in determining the steady-state state vector values.

To start with this consideration, remain the well known result (see e.g. [38] and references therein). The solution for the 1D differential equation

$$\dot{x}(t) = Ax(t) + Bu(t)$$

is given by

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau.$$

Goals of the considered control scheme are defined the same as before (see the introduction of this chapter).

Suppose that stability along the pass holds. Then as $k \to \infty$ it is expected that for $0 \le t < \alpha$, $x_{k+1}(t) = x_k(t)$, $y_{k+1}(t) = y_{ref}(t)$ and hence the steady state model becomes

$$\dot{x}_\infty(t) = Ax_\infty(t) + B_0y_{ref}(t) + Bu_\infty(t) + Ew(t), \tag{5.31}$$
$$y_{ref}(t) = Cx_\infty(t) + D_0y_{ref}(t) + Du_\infty(t) + Fw(t). \tag{5.32}$$

Then introduce the following residual variables

$$\begin{aligned}
\hat{y}_k(t) &= y_k(t) - y_{ref}(t), \\
\hat{x}_k(t) &= x_k(t) - x_\infty(t), \\
\hat{u}_k(t) &= u_k(t) - u_\infty(t).
\end{aligned}$$

Next, subtract (5.31)-(5.32) from (5.3)-(5.4) to obtain

$$\dot{\hat{x}}_{k+1}(t) = \widetilde{A}\hat{x}_{k+1}(t) + B\hat{u}_{k+1}(t) + \widetilde{B}_0\hat{y}_k(t),$$
$$\hat{y}_{k+1}(t) = \widetilde{C}\hat{x}_{k+1}(t) + D\hat{u}_{k+1}(t) + \widetilde{D}_0\hat{y}_k(t)$$

and in this so-called residual model the influence of the disturbance has been rejected. Now the application of the stabilizing control law is required. Then this can be achieved by considering the following control law (which is the residual version of (4.16))

$$\hat{u}_{k+1}(t) = K_1\hat{x}_{k+1}(t) + K_2\hat{y}_k(t)$$

or, equivalently using the original variables,

$$u_{k+1}(t) = u_\infty(t) + K_1 x_{k+1}(t) - K_1 x_\infty(t) + K_2 y_k(t) - K_2 y_{ref}(t) \tag{5.33}$$

and the task now is to construct $x_\infty(t)$ and $u_\infty(t)$, when $w(t)$ and $y_{ref}(t)$ are assumed to be known.

Note that equation of (5.32) is static hence assuming that there exists - the inverse of D (in a case when D is square and nonsingular) or - the pseudo-inverse denoted by $D^\sharp$ (in a case when D is not square or/and nonsingular) - it can be written as

$$u_\infty(t) = D^{-1}((I - D_0)y_{ref}(t) - Cx_\infty(t) - Fw(t)) \tag{5.34}$$

or

$$u_\infty(t) = D^\sharp((I - D_0)y_{ref}(t) - Cx_\infty(t) - Fw(t)).$$

Hence (5.31) becomes

$$\dot{x}_\infty(t) = (A - BD^{-1}C)x_\infty(t) + (B_0 + BD^{-1}(I - D_0))y_{ref}(t) + (E - BD^{-1}F)w(t), \tag{5.35}$$

which is the differential equation.

Define the matrices

$$\tilde{A} = A - BD^{-1}C, \;\; \tilde{B}_0 = B_0 + BD^{-1}(I - D_0), \;\; \tilde{E} = E - BD^{-1}F.$$

It is straightforward to see that the solution for (5.35) becomes

$$x_\infty(t) = e^{\tilde{A}t}x_\infty(0) + \tilde{B}_0\int_0^t e^{\tilde{A}(t-\tau)}y_{ref}(\tau)d\tau + \tilde{E}\int_0^t e^{\tilde{A}(t-\tau)}w(\tau)d\tau \tag{5.36}$$

and then, the input sequence $u_\infty(t)$ can be computed using (5.34)

Remark 5.3 *It is necessary to mention that there can arise serious numerical problems when attempting to determine the steady state components $x_\infty(t)$ and $u_\infty(t)$. This is due to the requirement of integration in (5.36), which acts like a 'bottleneck' here. It is also possible to compute $x_\infty(t)$ using the Laplace transform but, even that approach does not provide the required efficiency.*

Example 5.4 *To highlight the last result consider the case, when* $y_{ref}(t) = const \in R^m$ *(the simplest possible case) and* $w(t)$ *is an integrable function. Hence take* $y_{ref}(t) = -1$ *and disturbances as a full along the pass period of sine wave with the amplitude of* 0.5*, i.e.* $w(t) = 0.5sin(\frac{2\Pi t}{\alpha})$*. For this particular case (5.36) becomes*

$$\begin{aligned} x_\infty(t) =& e^{\tilde{A}t} - \frac{\tilde{B}_0\left(-1+e^{\tilde{A}t}\right)}{\tilde{A}} \\ & - \tilde{E}\alpha\left(2\pi\left(\cos(\frac{\pi t}{\alpha})\right)^2 - \pi + \tilde{A}\alpha\sin(\frac{\pi t}{\alpha})\cos(\frac{\pi t}{\alpha}) - \pi e^{\tilde{A}t}\right)\left(\tilde{A}^2\alpha^2 + 4\pi^2\right)^{-1}, \end{aligned}$$

which is the function of t *and hence it is possible to compute* $x_\infty(t)$ *for any specified* t*, however it is not trivial.*

Hence when $x_\infty(t)$ is provided, it is possible to compute $u_\infty(t)$. Note that since it is possible to apply the stabilization procedure for the residual model for computing controllers K_1 and K_2, all factors necessary to (5.33) have been provided.

5.3.4 Discrete LRPs – stability along the pass – output control

When the control scheme presented above is considered, it is strictly possible to swap the original state/output control law of (5.26) with the control law taking the advantage of the pass profiles (outputs) only sequence. It then becomes that (5.26) should be presented in the form similar to (4.47) or any extension of that control law (e.g. (4.54) or (4.64)).

Hence now it is assumed that the incremental model of (5.24)-(5.25) is given and disturbances have been already decoupled in this model.

Suppose again that a stabilizing control law is required. Then this can be achieved by rewriting (4.47) into the form appropriate for process of (5.24)-(5.25) which becomes as follows

$$\hat{u}_{k+1}(p) = \widetilde{K}_1\hat{y}_{k+1}(p) + \widetilde{K}_2\hat{y}_k(p)$$

or, equivalently,

$$u_{k+1}(p) = u_\infty(p) + \widetilde{K}_1 y_{k+1}(p) + \widetilde{K}_2 y_k(p) - (\widetilde{K}_1 + \widetilde{K}_2)y_{ref}(p), \tag{5.37}$$

since

$$\begin{cases} y_k(p) \\ y_{k+1}(p) \end{cases} \approx \quad y_{ref}(p) \text{ when } k \to +\infty.$$

The task now is to construct again $u_\infty(p)$ when $w(p)$ and $y_{ref}(p)$ are known. It can be done by the same operations as presented in the previous section. Hence apply (5.28) (or (5.29) for the square and nonsingular D) to compute $u_\infty(p)$ and (5.30) to compute $x_\infty(p)$.

In this case the overall control law is of the form (5.37), where $\widetilde{K}_1$ and $\widetilde{K}_2$ are the output control law matrices computed by the LMI method to give stability along the pass closed loop system.

It is straightforward to see that since $x_\infty(p)$ has been computed, it can be used backward to compute entries of $u_\infty(p),\ 0 \le p \le \alpha - 1$.

Remark 5.4 *As a possible drawback of the presented approach can be the fact that even, if the sequence of the steady state vectors $x_\infty(p)$ is not explicitly incorporated into the control law (5.37), it still has to be determined as necessary to compute $u_\infty(p)$. Hence the numerical efforts are increasing and, on the other hand, the necessity of additional computations can cause the numerical problems. Nevertheless, note that this step can be done off-line the control scheme (before the application of the control sequence), i.e. since (5.30) uses only known vectors ($y_{ref}(p)$, $w(t)$ and d_∞) and model matrices there are no apparent problems in computing it.*

Example 5.5 *Consider a case of (5.1)-(5.2) described by the following matrices*

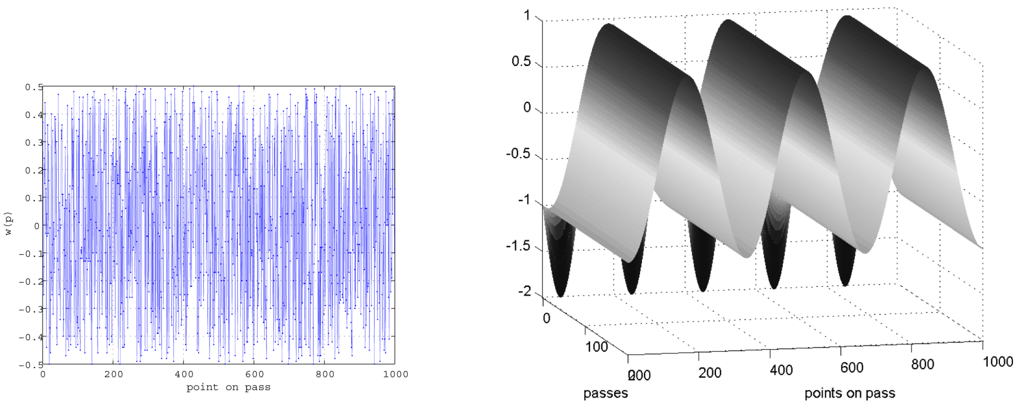
$$A = \begin{bmatrix} 0.8 & 0.19 & 0.6 \\ -0.08 & 0.21 & -0.73 \\ -0.63 & 0.56 & 0.78 \end{bmatrix},\ B = \begin{bmatrix} 2.5 \\ -1.45 \\ -0.76 \end{bmatrix},\ B_0 = \begin{bmatrix} 0 \\ 0.75 \\ -0.7 \end{bmatrix},\ E = \begin{bmatrix} -0.23 \\ 0.7 \\ 0.05 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.29 & -0.06 & 0.01 \end{bmatrix},\ D = \begin{bmatrix} 1.14 \end{bmatrix},\ D_0 = \begin{bmatrix} 1.09 \end{bmatrix},\ F = \begin{bmatrix} 0.61 \end{bmatrix}.$$

The pass length has been set as $\alpha = 1000$. The reference signal has been set to 3 full periods of cosine function $y_{ref}(p) = cos(\frac{-3\pi p}{0.5*\alpha} + 6\pi)$ over $0 \le p \le \alpha - 1$. The boundary conditions are $y_0(p) = -1 + sin(\frac{-5\pi p}{0.5*\alpha} + 10\pi)$ over $0 \le p \le \alpha - 1$ and $d_{k+1} = [1.8 \ \ -1.1 \ \ 0.43]^T$. The disturbances that influence the system have been set as random values over $0 \le p \le \alpha - 1$, generated with the following MATLAB *formula*

```
w(p)=round((rand()-0.5)*100)/100
```

and are shown in Figure 5.5 a)



a) b)

Figure 5.5. Disturbances (a) and pass profiles (b)

Due to the output controller design presented in Theorem 4.18 the following controllers have been computed

$$\widetilde{K}_1 = 6.4319,\ \widetilde{K}_2 = -4.8532.$$

Figure 5.5 b) shows the output of the controlled system.

5.4 Proportional plus Integral control of LRPs

In this section, the goals defined in the introduction of the chapter are again considered. To remain those, the basic design task is the development of control laws, which give asymptotic stability or/and stability along the pass and completely reject the effects of the disturbance terms (the same as before) is considered. In this section the proportional plus integral control approach is assumed to be applied. For the classical 1D systems the similar control scheme has been considered, for the description refer to [150] and references therein.

5.4.1 Discrete LRPs – asymptotic stability – 1D equivalent model

Again, an obvious target is to force the closed loop process to produce a pre-specified pass profile vector, say

$$V_{ref} = \begin{bmatrix} y_{ref}(0)^T & y_{ref}(1)^T & \dots & y_{ref}(\alpha-1)^T \end{bmatrix}^T$$

on a pre-specified pass. Also, this pass could be chosen as any finite value of k but here the request is that V_{ref} is the resulting limit profile.

If it is assumed that V_{ref} has only zero entries and there are no disturbances present, then asymptotic stability alone is enough to ensure this property. Here the more general case is considered when disturbances are present and also the target limit profile contains constant but non-zero entries. The method is to develop the scheme of proportional plus integral control for these processes in the pass-to-pass direction.

To begin, first note that the tracking error for any fixed value of l, say l^*, is given by $V(l^*) - V_{ref}$ and summing this over l passes gives the so-called total tracking error $R(l)$, i.e.

$$R(l) = \sum_{j=0}^{l}(V(l^*) - V_{ref})$$

and hence

$$R(l+1) = R(l) + V(l+1) - V_{ref}. \tag{5.38}$$

Now, introduce the so-called extended state vector, which incorporates the total tracking error, as

$$Z(l) = \begin{bmatrix} V(l) \\ R(l) \end{bmatrix}.$$

Then using (2.25) and (5.38), the evolution of $Z(l)$ is governed by

$$\begin{aligned} \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} Z(l+1) &= \begin{bmatrix} \Phi & 0 \\ 0 & I \end{bmatrix} Z(l) + \begin{bmatrix} \Delta \\ 0 \end{bmatrix} U(l) \\ &+ \begin{bmatrix} 0 \\ -I \end{bmatrix} V_{ref} + \begin{bmatrix} \Theta_0 \\ 0 \end{bmatrix} d_l + \begin{bmatrix} \Omega_y \\ 0 \end{bmatrix} W \end{aligned}$$

or

$$Z(l+1) = \begin{bmatrix} \Phi & 0 \\ \Phi & I \end{bmatrix} Z(l) + \begin{bmatrix} \Delta \\ \Delta \end{bmatrix} U(l) + \begin{bmatrix} 0 \\ -I \end{bmatrix} V_{ref} + \begin{bmatrix} \Theta_0 \\ \Theta_0 \end{bmatrix} d_l + \begin{bmatrix} \Omega_y \\ \Omega_y \end{bmatrix} W, \tag{5.39}$$

where

$$\Omega_y = \begin{bmatrix} F & 0 & 0 & \dots & 0 \\ CE & F & 0 & \dots & 0 \\ CAE & CE & F & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{\alpha-2}E & CA^{\alpha-3}E & CA^{\alpha-4}E & \dots & F \end{bmatrix}.$$

Suppose now that asymptotic stability holds. Then as $l \to \infty$

$$Z(l+1) = Z(l) \equiv Z_\infty$$

and also from (5.39)

$$Z_\infty = \begin{bmatrix} \Phi & 0 \\ \Phi & I \end{bmatrix} Z_\infty + \begin{bmatrix} \Delta \\ \Delta \end{bmatrix} U_\infty + \begin{bmatrix} 0 \\ -I \end{bmatrix} V_{ref} + \begin{bmatrix} \Theta_0 \\ \Theta_0 \end{bmatrix} d_\infty + \begin{bmatrix} \Omega_y \\ \Omega_y \end{bmatrix} W, \tag{5.40}$$

where U_∞ denotes the input applied on the limit profile and d_∞ the known state initial vector for the limit profile.

Now introduce the so-called residual variables (the incremental vectors)

$$\begin{aligned} \hat{Z}(l) &= Z(l) - Z_\infty, & \hat{U}(l) &= U(l) - U_\infty, \\ \hat{V}(l) &= V(l) - V_{ref}, & \hat{R}(l) &= R(l) - R_\infty, \\ \hat{d}_l &= d_l - d_\infty, \end{aligned}$$

and subtract (5.40) from (5.39) to obtain the incremental process dynamics as

$$\begin{aligned} \hat{Z}(l+1) &= \begin{bmatrix} \Phi & 0 \\ \Phi & I \end{bmatrix} \hat{Z}(l) + \begin{bmatrix} \Delta \\ \Delta \end{bmatrix} \hat{U}(l) + \begin{bmatrix} \Theta_0 \\ \Theta_0 \end{bmatrix} \hat{d}_l \\ &= \hat{\Phi}\hat{Z}(l) + \hat{\Delta}\hat{U}(l) + \hat{\Theta}_0 \hat{d}_l. \end{aligned} \tag{5.41}$$

This last linear system is unstable as a 1D equivalent model, due to the structure of the matrix $\hat{\Phi}$ but the external disturbance vector W has been completely decoupled. Hence if it is possible to make this model asymptotically stable, then both: the disturbance rejection and also the required limit profile are ensured. Consider, therefore, the following control law for (5.41)

$$\hat{U}(l) = \hat{K}\hat{Z}(l) = \begin{bmatrix} \hat{K}_1 & \hat{K}_2 \end{bmatrix} \begin{bmatrix} \hat{V}(l) \\ \hat{R}(l) \end{bmatrix} \tag{5.42}$$

or, equivalently,

$$U(l) = \hat{K}\big(Z(l) - Z_\infty\big) + U_\infty = \begin{bmatrix} \hat{K}_1 & \hat{K}_2 \end{bmatrix} \begin{bmatrix} V(l) - V_{ref} \\ R(l) - R_\infty \end{bmatrix} + U_\infty, \tag{5.43}$$

i.e. proportional plus integral action, where R_∞ denotes limit as $l \to \infty$ of $R(l)$.

To implement the control law of (5.43), it is clearly required to have R_∞ and U_∞. To provide these, first note that the second block row in (5.40) can be rewritten as

$$(I - \Phi)V_{ref} - \Theta_0 d_\infty - \Omega_y W = \Delta U_\infty \tag{5.44}$$

and hence (when letting $l \to \infty$ in (5.42)), it is straightforward to note that

$$U_\infty - \hat{K}_1 V_{ref} - \hat{K}_2 R_\infty = 0. \tag{5.45}$$

Now substitute (5.45) into (5.44) to obtain

$$[I - \Phi - \Delta\hat{K}_1]V_{ref} - \Theta_0 d_\infty - \Omega_y W - \Delta\hat{K}_2 R_\infty = 0$$

and this last expression gives the condition which must hold to achieve V_{ref} under the control law (5.43), i.e.

$$\text{rank}\,\big[I - \Phi - \Delta\hat{K}_1\big] = \text{rank}\,\big[I - \Phi - \Delta\hat{K}_1 \,|\, \Theta_0 d_\infty + \Omega_y W + \Delta\hat{K}_2 R_\infty\big]. \tag{5.46}$$

Note also that, if $\hat{K}_1$ stabilizes Φ, i.e. ensures that $r(\Phi + \Delta\hat{K}_1) < 1$, then $\big[I - \Phi - \Delta\hat{K}_1\big]$ is invertible and this last condition always holds.

Suppose now that (5.46) holds, then it can be used to compute R_∞ and hence, using (5.45), U_∞. Moreover, using (5.45), it is possible to rewrite the proportional plus integral control law (5.43) as

$$U(l) = \hat{K}_1 V(l) + \hat{K}_2 R(l) = \hat{K} Z(l) \tag{5.47}$$

and the controller $\hat{K}$ here can, in effect, be computed using Theorem 4.3 or Theorem 4.4 since (5.41) is a 1D state equation with $\hat{\Phi}$ as a system matrix and $\hat{\Delta}$ as an input matrix. Then (5.47) is in the form of a 1D state feedback control law. This leads to the main design result.

Theorem 5.1 *Suppose that a discrete LRP described by (5.1)-(5.2), given in the form of (5.41) is subject to the control law which can be written in the form (5.47). Then the resulting closed loop process is asymptotically stable with prescribed limit profile V_{ref} if there exist matrices $\hat{P} > 0$, $\hat{G}$, $\hat{L}$ such that the following LMI condition holds*

$$\begin{bmatrix} -\hat{P} & \hat{\Phi}\hat{G} + \hat{\Delta}\hat{L} \\ \hat{G}^T\hat{\Phi}^T + \hat{L}^T\hat{\Delta} & \hat{P} - \hat{G} - \hat{G}^T \end{bmatrix} < 0.$$

Also if this condition holds, then the controller matrix in (5.42) or (5.43) is given by

$$\hat{K} = \hat{L}\hat{G}^{-1}. \tag{5.48}$$

Proof. This follows identical steps to that of Theorem 4.4 and hence the details are omitted here. □

Example 5.6 *As a numerical example, consider the following case of (5.1)-(5.2)*

$$A = \begin{bmatrix} 0.23 & 0.36 & 0.37 \\ 0.29 & 0.14 & 0.10 \\ 0.74 & 0.39 & 0.69 \end{bmatrix},\ B = \begin{bmatrix} 3.12 \\ -4.89 \\ 5.77 \end{bmatrix},\ B_0 = \begin{bmatrix} 0.29 \\ 0.28 \\ 0.17 \end{bmatrix}, E = \begin{bmatrix} 0.19 \\ 0.04 \\ 0.1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.17 & 0.48 & 0.38 \end{bmatrix},\ D = 1.29,\ D_0 = -1, F = 0.2$$

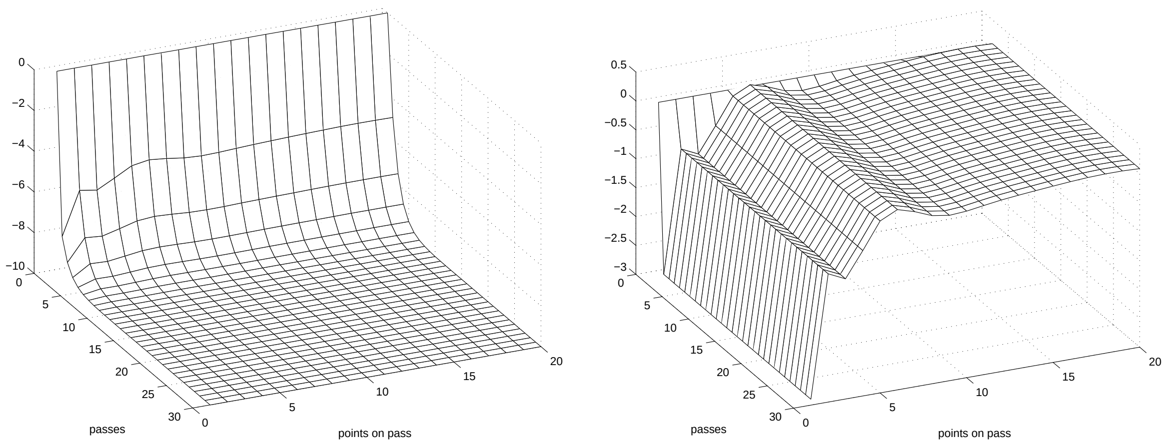
with $\alpha = 20$. Note that the considered model is asymptotically unstable since $r(D_0) = 1$. The boundary conditions are provided in the static form $d_{k+1} = [\,1.83\ \ -4.1\ \ -4.65\,]^T$, $k \geq 0$ and $y_0(p) = 0,\ 0 \leq p \leq \alpha - 1$. For the purpose of simulation the disturbance signal has been assumed as which denotes the sampled full period of the sine wave with the amplitude of -1 along the pass. The reference signal has been chosen as $y_{ref}(p) = -10,\ 0 \leq p \leq \alpha - 1$. The application of Theorem 5.1 provides the following controllers

$$\hat{K}_1 = \begin{bmatrix} \hat{K}_1^1 & \hat{K}_1^2 & \hat{K}_1^3 & \dots & \hat{K}_1^{20} \end{bmatrix} = \begin{bmatrix} \hat{k}_1^1 & 0 & 0 & \dots & 0 \\ \hat{k}_1^2 & \hat{k}_1^1 & 0 & \dots & 0 \\ \hat{k}_1^3 & \hat{k}_1^2 & \hat{k}_1^1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{k}_1^{20} & \hat{k}_1^{19} & \hat{k}_1^{18} & \dots & \hat{k}_1^1 \end{bmatrix}, \quad \hat{K}_2 = \begin{bmatrix} \hat{K}_2^1 & \hat{K}_2^2 & \hat{K}_2^3 & \dots & \hat{K}_2^{20} \end{bmatrix} = \begin{bmatrix} \hat{k}_2^1 & 0 & 0 & \dots & 0 \\ \hat{k}_2^2 & \hat{k}_2^1 & 0 & \dots & 0 \\ \hat{k}_2^3 & \hat{k}_2^2 & \hat{k}_2^1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{k}_2^{20} & \hat{k}_2^{19} & \hat{k}_2^{18} & \dots & \hat{k}_2^1 \end{bmatrix},$$

where

$$\begin{aligned} \hat{K}_1^1 &= \Big[\ 0.77 \;\; 0.50 \;\; 1.42 \;\; 0.90 \;\; 2.05 \;\; -1.62 \;\; 2.99 \;\; -2.79 \;\; 4.46 \;\; -4.62 \\ &\quad 6.75 \;\; -7.49 \;\; 10.36 \;\; -12.01 \;\; 16.02 \;\; -19.01 \;\; 24.90 \;\; -30.22 \;\; 38.84 \;\; -47.69\ \Big]^T, \\ \hat{K}_2^1 &= \Big[\ 0.39 \;\; 0.21 \;\; 0.66 \;\; 0.38 \;\; 0.94 \;\; 0.70 \;\; -1.37 \;\; 1.22 \;\; -2.02 \;\; 2.04 \\ &\quad -3.04 \;\; 3.32 \;\; -4.66 \;\; 5.34 \;\; -7.18 \;\; 8.50 \;\; -11.15 \;\; 13.48 \;\; -17.38 \;\; 21.28\ \Big]^T. \end{aligned}$$

To present the usefulness of the considered approach the set of tests has been performed. The following figures show the output of the considered process with controllers computed according to Theorem 5.1 (Figure 5.6 a)) and (to compare the result and prove the applicability) Theorem 4.4 (Figure 5.6 b)) applied.



a) The output with controller (5.47) applied

b) The output with controller computed according to Theorem 4.4 applied

Figure 5.6. Pass profiles produced under varied sets of controllers applied

It is straightforward to see that only application of the complete control law of (5.47) provides the disturbances rejection, required output value and the appropriate dynamics. These are shown in Figure 5.6 a).In comparison, Figure 5.6 b) presents the same process, with controllers computed according to Theorem 4.4 (the control law of (4.3) applied). On the other hand, Figure 5.7 presents the output with only $\hat{K}_1$ *of controller (5.47) applied. The applicability of the integral part in the control law is straightforward to see if refer to Figure 5.7. The integral part of the control law in this case ensures the acceptable performance of the closed loop system.*

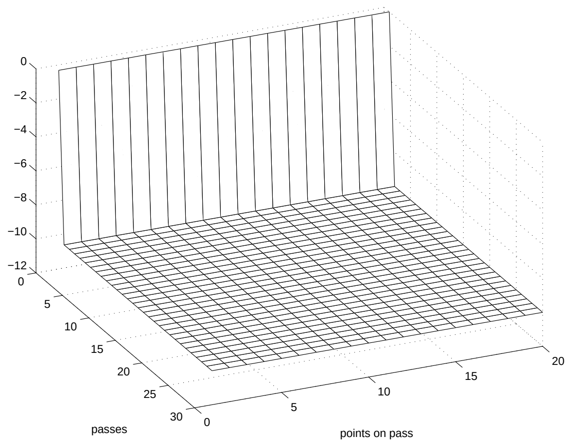


Figure 5.7. The output with only $\hat{K}_1$ of controller (5.47) applied

5.4.2 Discrete LRPs – stability along the pass

In terms of ‘acceptable, or desired, performance from a given example, in general the stronger demand regarding the stability is ensuring stability along the pass — this guarantees the existence of a limit profile which is stable as a 1D discrete linear system. The problem, how to ensure stability along the pass for examples have been presented in Section 4.4. In particular, it has been shown that a control law of the form (4.11) can be used to give this property. Moreover, the design of the control law matrices can be implemented using LMIs where the basic result starts from interpreting Theorem 4.5 for the resulting closed loop state-space model.

Again, the question stated here is how to obtain a specified limit profile $y_{ref}(p)$ in the presence of disturbances.

Consider the state-space model (5.1)-(5.2) at the point p on the pass k. Then the total tracking error at this point is defined as

$$\chi_k(p) := \sum_{j=0}^{k} \Big(y_j(p) - y_{ref}(p) \Big),$$

i.e. the error at the point p summed from pass 0 to k. Substitution from the process state-space model now gives

$$\begin{aligned} \chi_{k+1}(p) &= \chi_k(p) + y_{k+1}(p) - y_{ref}(p) \\ &= \chi_k(p) + Cx_{k+1}(p) + D_0 y_k(p) + Du_{k+1}(p) + Fw(p) - y_{ref}(p). \end{aligned} \tag{5.49}$$

Now, introduce the so-called extended output (pass profile) vector

$$z_{k+1}(p) := \begin{bmatrix} y_{k+1}(p) \\ \chi_{k+1}(p) \end{bmatrix}.$$

Then (5.49) yields

$$z_{k+1}(p) = \begin{bmatrix} C \\ C \end{bmatrix} x_{k+1}(p) + \begin{bmatrix} D_0 & 0 \\ D_0 & I \end{bmatrix} z_k(p) + \begin{bmatrix} D \\ D \end{bmatrix} u_{k+1}(p) + \begin{bmatrix} 0 \\ -I \end{bmatrix} y_{ref}(p) + \begin{bmatrix} F \\ F \end{bmatrix} w(p).$$

Suppose now that the process of (5.1)-(5.2) is asymptotically stable, i.e. $r(D_0) < 1$, then as $k \to \infty$

$$x_{k+1}(p) = x_k(p) \equiv x_\infty(p),$$
$$y_{k+1}(p) = y_k(p) \equiv y_\infty(p),$$
$$u_{k+1}(p) = u_k(p) \equiv u_\infty(p)$$

and let $\chi_\infty(p)$ denote $\lim_{k\to\infty} \chi_k(p)$. Then it can be written

$$\chi_{k+1}(p) = \chi_k(p) \equiv \chi_\infty(p)$$

and hence

$$x_\infty(p+1) = Ax_\infty(p) + B_0 y_\infty(p) + Bu_\infty(p) + Ew(p), \tag{5.50}$$

$$z_\infty(p) = \begin{bmatrix} C \\ C \end{bmatrix} x_\infty(p) + \begin{bmatrix} D_0 & 0 \\ D_0 & I \end{bmatrix} z_s(p) + \begin{bmatrix} D \\ D \end{bmatrix} u_s(p) + \begin{bmatrix} 0 \\ -I \end{bmatrix} y_{ref}(p) + \begin{bmatrix} F \\ F \end{bmatrix} w(p), \tag{5.51}$$

where $z_\infty(p) = \lim_{k\to\infty} z_k(p)$.

Next, define the following so-called incremental vectors

$$\begin{aligned} \hat{z}_k(p) &= z_k(p) - z_\infty(p), \\ \hat{u}_k(p) &= u_k(p) - u_\infty(p), \\ \hat{x}_k(p) &= x_k(p) - x_\infty(p). \end{aligned}$$

Then using (5.1)-(5.2) and (5.50)-(5.51), it is straightforward to obtain

$$\hat{x}_{k+1}(p+1) = A\hat{x}_{k+1}(p) + \hat{B}_0\hat{z}_k(p) + B\hat{u}_{k+1}(p), \tag{5.52}$$
$$\hat{z}_{k+1}(p) = \hat{C}\hat{x}_{k+1}(p) + \hat{D}_0\hat{z}_k(p) + \hat{D}\hat{u}_{k+1}(p), \tag{5.53}$$

where

$$\hat{B}_0 = \begin{bmatrix} B_0 & 0 \end{bmatrix}, \ \hat{C} = \begin{bmatrix} C \\ C \end{bmatrix}, \ \hat{D}_0 = \begin{bmatrix} D_0 & 0 \\ D_0 & I \end{bmatrix}, \ \hat{D} = \begin{bmatrix} D \\ D \end{bmatrix}$$

and the key point now is that the influence of the disturbance has been completely rejected. The task now is to meet the specification, when the limit profile (for the original process) be equal to the prescribed vector $y_{ref}(p)$. Note that (5.52)-(5.53) is of the structure of discrete LRP of (2.9)-(2.10) and hence known (formerly presented) methods for the synthesis can be applied.

What is more, the matrix $\hat{D}_0$ in (5.53) always has eigenvalues with modulus at least equal to unity and hence this discrete LRP state-space model is asymptotically unstable and hence unstable along the pass. To obtain any (and in particular the required) limit profile from it, the control action must be applied. Moreover, in order to make this limit profile equal to $y_{ref}(p)$ with the specified 1D transient performance specifications, stability along the pass in the closed loop is also required.

Now consider applying the control law, which is of the form (4.11) but applied for the extended model (5.52)-(5.53)

$$\begin{aligned} \hat{u}_{k+1}(p) &= K_x\hat{x}_{k+1}(p) + K_z\hat{z}_k(p) \\ &= K_x\hat{x}_{k+1}(p) + K_{z1}\hat{y}_k(p) + K_{z2}\hat{\chi}_k(p) \\ &= \begin{bmatrix} K_x & K_{z1} & K_{z2} \end{bmatrix} \begin{bmatrix} \hat{x}_{k+1}(p) \\ \hat{y}_k(p) \\ \hat{\chi}_k(p) \end{bmatrix}. \end{aligned} \tag{5.54}$$

Then the following result gives an LMI based sufficient condition for closed loop stability along the pass together with a formula for computing the control law matrices. The proof of this result follows immediately on interpreting Theorem (4.9) and hence the details are omitted here.

Theorem 5.2 *Suppose that a control law of the form (5.54) is applied to a discrete LRP described by a state-space model of the form (5.52)-(5.53). Then the resulting closed loop process is stable along the pass if there exist matrices* $Y > 0$, $Z > 0$, *and* N *such that the following LMI holds*

$$\begin{bmatrix} Z - Y & 0 & Y\widetilde{A}_1^T + N^T\widetilde{B}_1^T \\ 0 & -Z & Y\widetilde{A}_2^T + N^T\widetilde{B}_2^T \\ \widetilde{A}_1 Y + \widetilde{B}_1 N & \widetilde{A}_2 Y + \widetilde{B}_2 N & -Y \end{bmatrix} < 0,$$

where

$$\widetilde{A}_1 = \begin{bmatrix} A & \hat{B}_0 \\ 0 & 0 \end{bmatrix},\ \widetilde{A}_2 = \begin{bmatrix} 0 & 0 \\ \hat{C} & \hat{D}_0 \end{bmatrix},\ \widetilde{B}_1 = \begin{bmatrix} B \\ 0 \end{bmatrix},\ \widetilde{B}_2 = \begin{bmatrix} 0 \\ \hat{D} \end{bmatrix}.$$

If this condition holds, then the matrices in the control law are given by

$$[K_x\ K_{z1}\ K_{z2}] = NY^{-1}$$

Suppose now that this last result holds. Then it follows immediately that $y_\infty(p) = y_{ref}(p)$ as required. Moreover

$$\begin{aligned} u_{k+1}(p) &= K_x\Big(x_{k+1}(p) - x_\infty(p)\Big) + K_{z1}\Big(y_k(p) - y_{ref}(p)\Big) \\ &+ K_{z2}\Big(\chi_k(p) - \chi_\infty(p)\Big) + u_\infty(p) \end{aligned}$$

and also

$$-K_x x_\infty(p) - K_{z1} y_{ref}(p) - K_{z2}\chi_\infty(p) + u_\infty(p) = 0.$$

Hence the final form of the control law to be applied to the original process is

$$u_{k+1}(p) = K_x x_{k+1}(p) + K_{z1} y_k(p) + K_{z2}\chi_k(p). \tag{5.55}$$

Rewrite now the part of the right-hand side of the control law (5.54) in the form

$$\begin{aligned} K_x x_{k+1}(p) + K_{z1} y_k(p) &= \begin{bmatrix} K_x & K_{z1} \end{bmatrix} \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \\ &=: KX^a_{k+1}(p), \end{aligned}$$

where $X^a_{k+1}(p)$ is termed the augmented state vector. Then immediately, it is to see that the final control law (5.55) has a two term structure, where the first term $KX^a_{k+1}(p)$ is static (proportional control action for stability) and the second $K_{z2}\chi_k(p)$ is the integral action to enforce the tracking of the requested limit profile $y_{ref}(p)$.

Example 5.7 *Consider the case of (5.1)-(5.2) given by*

$$A = \begin{bmatrix} 0.92 & 0.14 & -0.98 & 0.41 \\ -0.76 & -0.93 & -0.62 & 0.13 \\ 0.68 & -0.65 & 1.02 & -0.81 \\ 0.94 & 0.04 & 0.83 & 0.2 \end{bmatrix},\ B = \begin{bmatrix} 0.99 & -0.99 & 0.07 \\ 0.07 & -0.94 & -0.63 \\ 0.98 & -0.73 & 0.02 \\ -0.37 & 0.19 & -0.65 \end{bmatrix},\ E = \begin{bmatrix} 0.91 \\ 0.59 \\ 0.18 \\ 0.31 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} -0.01 & -0.43 \\ 0.29 & -0.13 \\ 0.98 & 1.09 \\ 1.09 & 0.17 \end{bmatrix}, \; C = \begin{bmatrix} -0.75 & 0.75 & 0.31 & 0.84 \\ -0.86 & 0.99 & 0.33 & -0.84 \end{bmatrix},$$

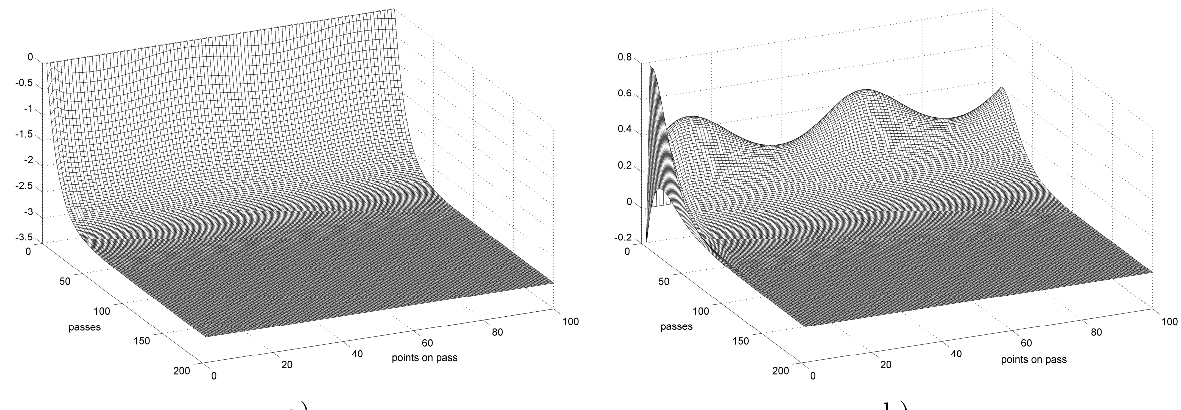
$$D = \begin{bmatrix} -0.33 & -0.14 & 0.59 \\ -0.18 & 0.94 & -0.17 \end{bmatrix}, \; D_0 = \begin{bmatrix} 1.11 & -0.66 \\ 0.46 & 1.23 \end{bmatrix}, \; F = \begin{bmatrix} -0.06 \\ 0.36 \end{bmatrix},$$

over the pass length $\alpha = 100$. The disturbances $w(p)$ are two full periods of sine wave. This example is asymptotically unstable, and hence unstable along the pass, since $r(D_0) = 1.2919$. The application of Theorem 5.2 provides the following controller matrices

$$K_x = \begin{bmatrix} -1.7945 & 1.0158 & 0.134 & 1.1219 \\ 0.5075 & -0.5836 & -0.5438 & 1.1854 \\ 0.3834 & -0.7341 & -0.6099 & -0.4623 \end{bmatrix},$$

$$K_{z1} = \begin{bmatrix} 1.8756 & -0.6793 \\ 0.0952 & -1.3957 \\ -0.7203 & 0.3862 \end{bmatrix}, \; K_{z2} = \begin{bmatrix} 0.0673 & 0.0407 \\ 0.0441 & -0.0559 \\ -0.0606 & 0.0056 \end{bmatrix}.$$

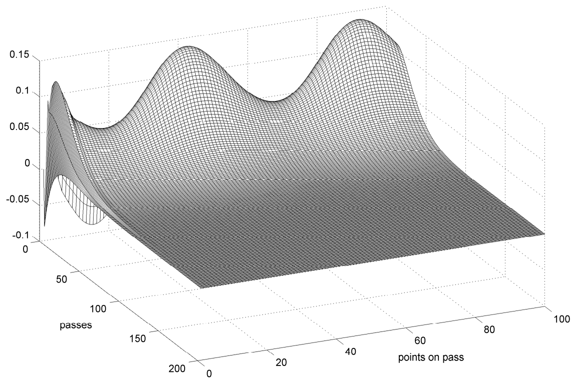
In order to accomplish the requested quality of the closed loop performance, the emphasis is put on the fact that the limit profile is a 1D discrete linear system and follows the standard route of using a step signal applied in each of two channels in turn. Figure 5.8 shows the closed loop responses to the case when $y_{ref}(p) = [\, -3 \; 0 \,]^T$, $0 \le p \le 99$. Here, the interaction in the second channel is relatively large at the beginning, but it is to note that the process converges quickly to the limit profile which has exactly the along the pass dynamics predicted and, in particular, the integral term completely removes the interaction. Figure 5.9 shows the closed loop responses in the case when $y_{ref}(p) = [\, 0 \; 3 \,]^T$, $0 \le p \le 99$, and the same comments hold. (These simulations are made for the following boundary conditions $d_{k+1} = [\, -1.33 \; -0.32 \; 1.13 \; -1.57 \,]^T$ and initial profile $y_0(p) = 0$, $0 \le p \le \alpha - 1$.)



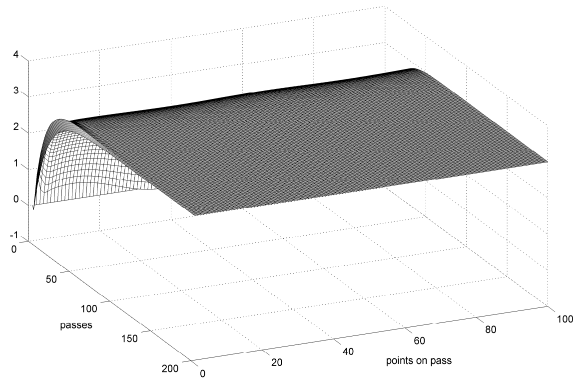
a) b)

Figure 5.8. First (a) and second (b) channel responses to $[-3 \;\; 0]^T$

Note that in both presented cases, there is an influence of disturbances visible on the presented figures. However, it vanishes when the evolution of the system continues, i.e. the disturbance effect is decoupled from the output.



a)



b)

Figure 5.9. First (a) and second (b) channel responses to $[0\ \ 3]^T$

5.4.3 Differential LRPs – stability along the pass

The proportional plus integral control scheme can be also provided for differential LRP of (5.3)-(5.4). Still, three main control goals defined before are to be meet. The whole following procedure follows the discrete case.

First, for the pass k and the position $t:\ 0 \leq t < \alpha$ along this pass define the total tracking error $\chi_k(t)$ as

$$\chi_k(t) = \sum_{j=0}^{k} \big(y_j(t) - y_{ref}(t)\big).$$

Then it follows immediately that

$$\chi_{k+1}(t) = \chi_k(t) + y_{k+1}(t) - y_{ref}(t)$$

or, using (5.4),

$$\chi_{k+1}(t) = \chi_k(t) + Cx_{k+1}(t) + Du_{k+1}(t) + D_0 y_k(t) + Fw(t) - y_{ref}(t). \tag{5.56}$$

Also, introduce the so-called extended pass profile vector as

$$z_k(t) = \begin{bmatrix} y_k(t) \\ \chi_k(t) \end{bmatrix}.$$

Then use of (5.4) together with (5.56) yields the following state-space model of the so-called augmented linear repetitive process

$$\begin{aligned} \dot{x}_{k+1}(t) &= Ax_{k+1}(t) + [B_0\ \ 0\,]\, z_k(t) + Bu_{k+1}(t) + Ew(t), & (5.57)\\ z_{k+1}(t) &= \begin{bmatrix} C \\ C \end{bmatrix} x_{k+1}(t) + \begin{bmatrix} D_0 & 0 \\ D_0 & I \end{bmatrix} z_k(t) + \begin{bmatrix} 0 \\ -I \end{bmatrix} y_{ref}(t) & (5.58)\\ &\quad + \begin{bmatrix} D \\ D \end{bmatrix} u_{k+1}(t) + \begin{bmatrix} F \\ F \end{bmatrix} w(t). \end{aligned}$$

Suppose that as $k \to \infty$, $x_k(t) \to x_\infty(t)$, $u_k(t) \to u_\infty(t)$ and $y_k(t) \to y_{ref}(t)$, $\chi_k(t) \to \chi_\infty(t)$, (hence $z_k(t) \to z_\infty(t)$). Then from (5.57)-(5.58), it is straightforward to obtain

$$\begin{aligned}
\dot{x}_\infty(t) &= Ax_\infty(t) + [B_0 \;\; 0\,]\, z_\infty(t) + Bu_\infty(t) + Ew(t), & (5.59)\\
z_\infty(t) &= \begin{bmatrix} C \\ C \end{bmatrix} x_\infty(t) + \begin{bmatrix} D_0 & 0 \\ D_0 & I \end{bmatrix} z_\infty(t) + & (5.60)\\
&\begin{bmatrix} D \\ D \end{bmatrix} u_\infty(t) + \begin{bmatrix} 0 \\ -I \end{bmatrix} y_{ref}(t) + \begin{bmatrix} F \\ F \end{bmatrix} w(t).
\end{aligned}$$

Now define the following incremental vectors

$$\begin{aligned}
\hat{z}_k(t) &= z_k(t) - z_\infty(t), & (5.61)\\
\hat{u}_k(t) &= u_k(t) - u_\infty(t),\\
\hat{x}_k(t) &= x_k(t) - x_\infty(t).
\end{aligned}$$

Then subtracting (5.59)-(5.60) from (5.57)-(5.58) and using (5.61) yields

$$\begin{aligned}
\dot{\hat{x}}_{k+1}(t) &= A\hat{x}_{k+1}(t) + \hat{B}_0\hat{z}_k(t) + B\hat{u}_{k+1}(t), & (5.62)\\
\hat{z}_{k+1}(t) &= \hat{C}\hat{x}_{k+1}(t) + \hat{D}_0\hat{z}_k(t) + \hat{D}\hat{u}_{k+1}(t), & (5.63)
\end{aligned}$$

where

$$\hat{B}_0 = \begin{bmatrix} B_0 & 0 \end{bmatrix},\; \hat{C} = \begin{bmatrix} C \\ C \end{bmatrix},\; \hat{D}_0 = \begin{bmatrix} D_0 & 0 \\ D_0 & I \end{bmatrix},\; \hat{D} = \begin{bmatrix} D \\ D \end{bmatrix}$$

and hence the disturbance term $w(t)$ is completely decoupled from the process dynamics. The only problem in the above analysis is that (5.62)-(5.63) is asymptotically unstable (this property is determined by the eigenvalues of the matrix $\hat{D}_0$ and some of these are equal to unity) and hence unstable along the pass. Consequently, the result is only achievable, if it is possible to find a control law to govern this property.

Next, consider the control law defined by

$$\begin{aligned}
\hat{u}_{k+1}(t) &= K_x\hat{x}_{k+1}(t) + K_z\hat{z}_k(t) & (5.64)\\
&= K_x\hat{x}_{k+1}(t) + K_{z1}\hat{y}_k(t) + K_{z2}\hat{\chi}_k(t)\\
&= \begin{bmatrix} K_x & K_{z1} & K_{z2} \end{bmatrix} \begin{bmatrix} \hat{x}_{k+1}(t) \\ \hat{y}_k(t) \\ \hat{\chi}_k(t) \end{bmatrix},
\end{aligned}$$

which is the differential LRP version of the classical proportional plus integral control action. (In the case of the integral action, this arises from the total tracking error contribution which is formed by summing across the passes.)

Now the following result which shows how to design this control law to ensure that (5.52) is stable along the pass can be presented

Theorem 5.3 *Suppose that the model of (5.62)-(5.63) is a subject to the control law of the form (5.64). Then the resulting closed loop process is stable along the pass if there exist matrices* $\hat{Y} > 0$, $\hat{Z} > 0$, $\hat{M}$ *and* $\hat{N}$ *such that the following LMI holds*

$$\begin{bmatrix} \hat{Y}A^T + A\hat{Y} + \hat{N}^T B^T + B\hat{N} & \hat{B}_0\hat{Z} + B\hat{M} & \hat{Y}\hat{C}^T + \hat{N}^T\hat{D}^T \\ \hat{Z}\hat{B}_0^T + \hat{M}^T B^T & -\hat{Z} & \hat{Z}\hat{D}_0^T + \hat{M}^T\hat{D}^T \\ \hat{C}\hat{Y} + \hat{D}\hat{N} & \hat{D}_0\hat{Z} + \hat{D}\hat{M} & -\hat{Z} \end{bmatrix} < 0. \quad (5.65)$$

If this condition holds, the control law matrices K_x and K_z are given by

$$K_x = \hat{N}\hat{Y}^{-1}, \;\; K_z = \hat{M}\hat{Z}^{-1}. \tag{5.66}$$

Proof. Simply note that (5.62)-(5.63) is of the form (2.14)-(2.15) and hence Theorem 4.8 can be applied to the closed loop process state-space model. □

To show how (5.64) can be actually employed, note that

$$\begin{aligned} \hat{u}_{k+1}(t) &= u_{k+1}(t) - u_\infty(t) \\ &= K_x \hat{x}_{k+1}(t) + K_z \begin{bmatrix} y_k(t) - y_{ref}(t) \\ \chi_k(t) - \chi_\infty(t) \end{bmatrix} \end{aligned} \tag{5.67}$$

or, using the original variables,

$$u_{k+1}(t) = K_x\Big(x_{k+1}(t) - x_\infty(t)\Big) + K_{z1}\Big(y_k(t) - y_{ref}(t)\Big) + K_{z2}\Big(\chi_k(t) - \chi_\infty(t)\Big) + u_\infty(t). \tag{5.68}$$

This control law can also be applied to the process in non-incremental form, i.e. as

$$u_{k+1}(t) = K_x x_{k+1}(t) + K_{z1} y_k(t) + K_{z2}\chi_k(t). \tag{5.69}$$

Then from (5.67), it is straightforward to see that

$$-K_x x_\infty(t) - K_{z1} y_{ref}(t) - K_{z2}\chi_\infty(t) + u_\infty(t) = 0. \tag{5.70}$$

Consequently, on any pass, it is not required to know information, which is generated on future passes, i.e. $\chi_\infty(t)$ and $u_\infty(t)$, which considerably simplifies the effort required to construct the control law output to be applied to the process since there is no need to pre-compute these two terms.

Example 5.8 *Consider again the model of unstable (5.3)-(5.4), given in Example 4.5 with the pass length $\alpha = 50$. The matrices E and F are assumed as follows*

$$E = \begin{bmatrix} 1 \\ 0.9 \\ 0.6 \\ -0.1 \end{bmatrix}, \; F = \begin{bmatrix} -1.7 \\ 1.3 \end{bmatrix}.$$

The application of Theorem 5.3 provides the following matrices

$$\hat{Y} = \begin{bmatrix} 1671.4553 & 1312.04 & -106.7492 & 1358.7098 \\ 1312.04 & 3470.7294 & 64.9729 & 2383.0249 \\ -106.7492 & 64.9729 & 1096.1297 & -581.8855 \\ 1358.7098 & 2383.0249 & -581.8855 & 3086.9887 \end{bmatrix},$$

$$\hat{Z} = \begin{bmatrix} 1773.1376 & -38.3513 & -73.2392 & 187.5560 \\ -38.3513 & 2631.0986 & 58.8763 & -134.2619 \\ -73.2392 & 58.8763 & 2235.3449 & -12.3697 \\ 187.5560 & -134.2619 & -12.3697 & 2366.2669 \end{bmatrix},$$

$$\hat{N} = \begin{bmatrix} 1483.1204 & 3227.0565 & -20.1667 & 4680.0674 \\ -1172.5727 & -2407.0977 & -60.2941 & -3565.3655 \\ -112.0671 & -532.3344 & 6.6859 & -244.4569 \end{bmatrix},$$

$$\hat{M} = \begin{bmatrix} -165.9026 & -614.1769 & 65.9177 & -72.8102 \\ -98.6652 & 399.5965 & 38.4149 & -90.5024 \\ -156.2374 & -102.0499 & 83.3951 & 97.0102 \end{bmatrix}.$$

Hence the PI controllers computed due to (5.66) and applicable in (5.69) become

$$K_x = \begin{bmatrix} -0.5789 & -0.5339 & 1.2398 & 2.4167 \\ 0.3985 & 0.4791 & -1.0525 & -1.8986 \\ 0.0443 & -0.2430 & 0.08 & 0.1040 \end{bmatrix},$$

$$K_{z1} = \begin{bmatrix} -0.0935 & -0.2374 \\ -0.0492 & 0.1496 \\ -0.0924 & -0.0386 \end{bmatrix}, \; K_{z2} = \begin{bmatrix} 0.0325 & -0.0367 \\ 0.0115 & -0.0258 \\ 0.0356 & 0.0463 \end{bmatrix}.$$

The disturbance has been generated using the following MATLAB *code*

```
dww=20*pi/(alpha1-1);
W=2*cos(-10*pi:dww:10*pi)+(rand(1,alpha1)-0.5)/5 - 1;
```

and shown in Figure 5.10.

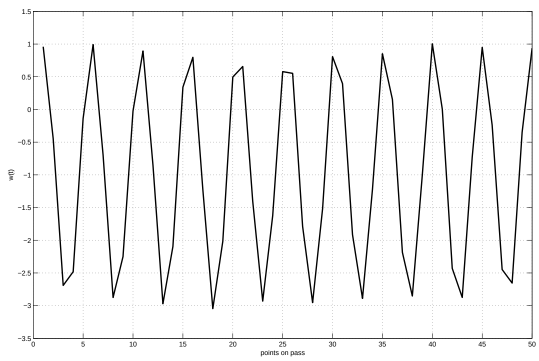


Figure 5.10. The disturbance $w(t)$

The initial pass profile is defined as follows $f(t) = [1 \;\; 1]^T, \;\; 0 \leq t < \alpha$ *and the initial state vectors* d_{k+1} *were chosen randomly at the beginning of each pass using the following formula*

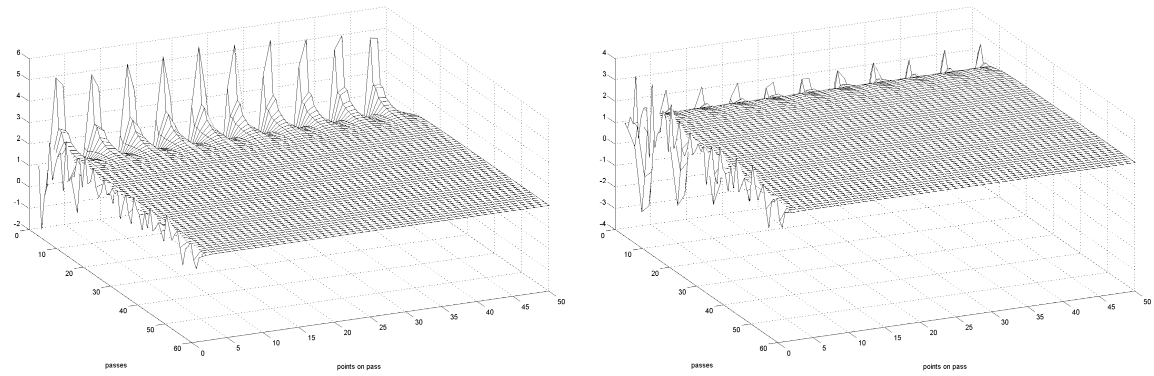
```
x0=round((rand(n,1)-0.5)*200)/100;
```

First, the reference signal is equal to $y_{ref}(t) = [2 \;\; 2]^T, \; 0 \leq t < \alpha$.

Figure 5.11 shows the responses of the resulting closed loop process.

This confirms that the design objectives have been satisfied, i.e. closed loop stability along the pass, the required limit profile is achieved and the influences of the disturbance have been rejected.

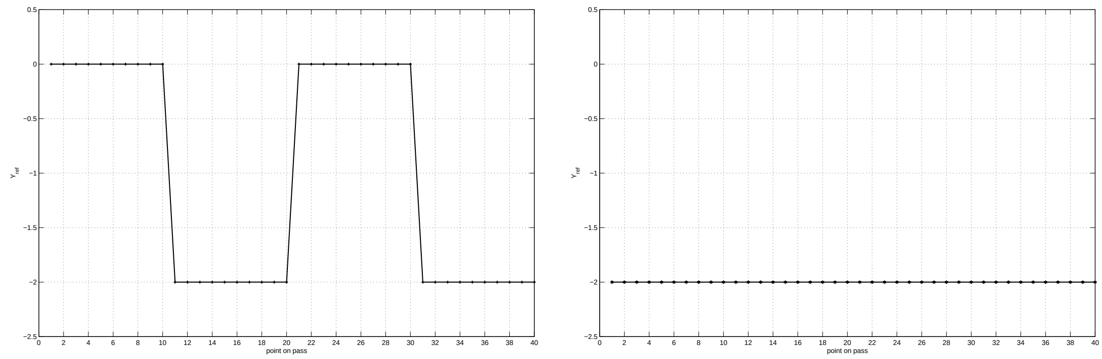
To highlight the proposed method abilities against different reference signals the following simulations were made (here $\alpha = 40$ *and* $d_{k+1} = [\,-0.22 \;\; -0.57 \;\; -0.81 \;\; -0.8\,]^T$*), where the*



Channel 1 Channel 2

Figure 5.11. The outputs of the closed loop repetitive process for the reference signals $[2\ 2]^T$

chosen reference signals are shown in Figure 5.12, Figure 5.14 and Figure 5.16 and the closed loop process pass profile (outputs) dynamics are shown in Figure 5.13, Figure 5.15 and Figure 5.17, respectively.



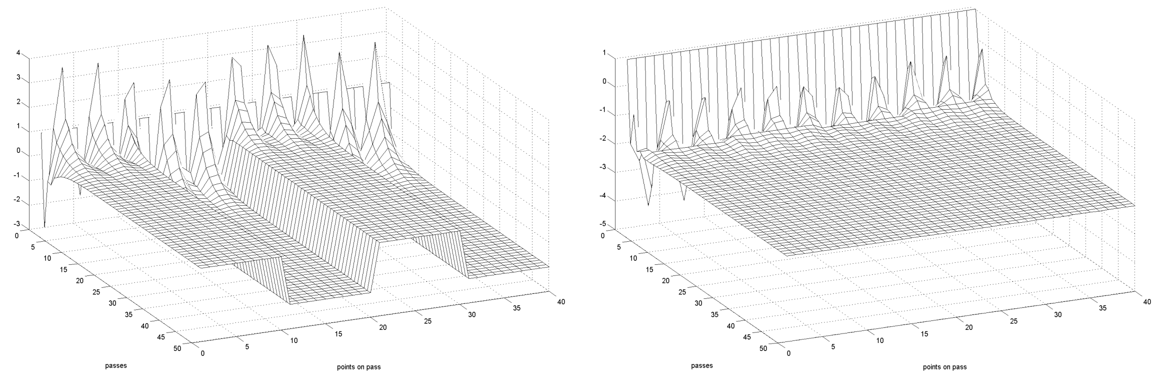
Channel 1 Channel 2

Figure 5.12. The reference signals

5.4.4 Discrete LRPs – stability along the pass – output control

By the analogy to Section 5.3.4, at this moment it is reasonable to ask about the possibility of application of the output control scheme to proportional plus integral approach. Hence assume that the incremental model of (5.52)-(5.53) is already given (i.e. the disturbances have been decoupled) and consider applying the following output control law

$$\begin{aligned} \hat{u}_{k+1}(p) &= \widetilde{K}_{z1}\hat{z}_{k+1}(p) + \widetilde{K}_{z2}\hat{z}_k(p) \\ &= \begin{bmatrix} \widetilde{K}_{y1} & \widetilde{K}_{\chi 1} & \widetilde{K}_{y2} & \widetilde{K}_{\chi 2} \end{bmatrix} \begin{bmatrix} \hat{y}_{k+1}(p) \\ \hat{\chi}_{k+1}(p) \\ \hat{y}_k(p) \\ \hat{\chi}_k(p) \end{bmatrix} \end{aligned} \tag{5.71}$$



Channel 1 Channel 2

Figure 5.13. Pass profiles for the reference signal shown in Figure 5.12

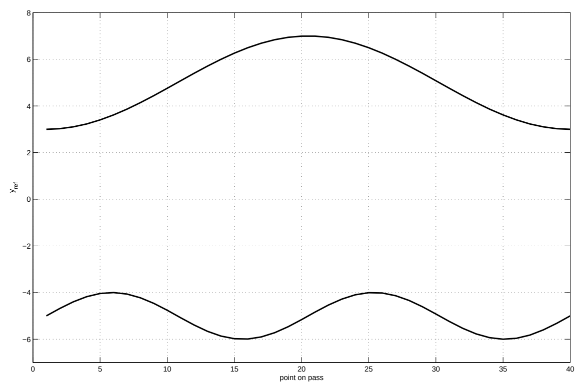


Figure 5.14. The reference signals

for that incremental model. Then the following result gives an LMI based sufficient condition for closed loop stability along the pass together with a formula for computing the control law matrices. The proof of this result follows immediately on interpreting Theorem 4.6 for the closed loop state-space model and hence the details are omitted here.

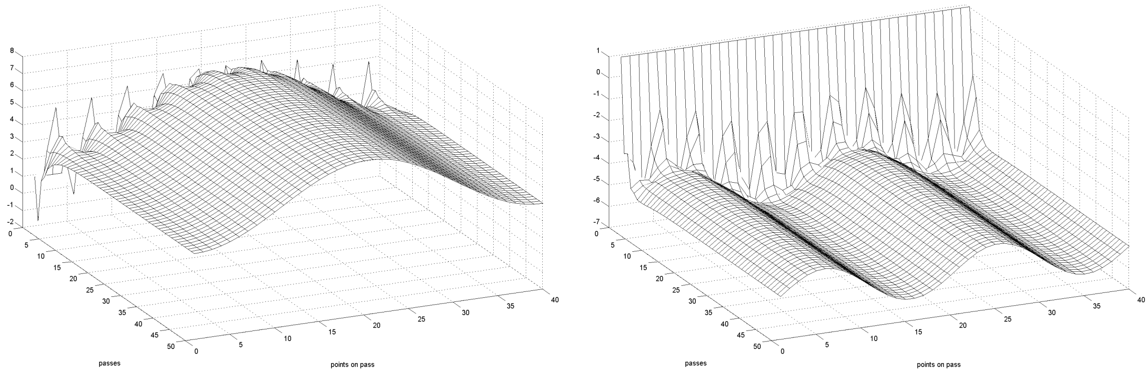
Theorem 5.4 *Suppose that a control law of the form (5.71) is applied to a discrete LRP described by a state-space model of the form (5.52)-(5.53). Then the resulting closed loop process is stable along the pass if there exist matrices* $Y > 0$, $Z > 0$, $X > 0$ *and* N *of appropriate dimensions such that the following LMI holds.*

$$\begin{bmatrix} Z - Y & (*) & (*) \\ 0 & -Z & (*) \\ \widetilde{A}_1 Y + \widetilde{B}_1 N \widetilde{C} & \widetilde{A}_2 Y + \widetilde{B}_2 N \widetilde{C} & -Y \end{bmatrix} < 0,$$

$$X\widetilde{C} = \widetilde{C}Y,$$

where

$$\widetilde{A}_1 = \begin{bmatrix} A & \hat{B}_0 \\ 0 & 0 \end{bmatrix}, \ \widetilde{A}_2 = \begin{bmatrix} 0 & 0 \\ \hat{C} & \hat{D}_0 \end{bmatrix}, \ \widetilde{B}_1 = \begin{bmatrix} B \\ 0 \end{bmatrix}, \ \widetilde{B}_2 = \begin{bmatrix} 0 \\ \hat{D} \end{bmatrix}, \ \widetilde{C} = \ diag(\hat{C}, I).$$



Channel 1 Channel 2

Figure 5.15. Pass profiles for the reference signal shown in Figure 5.14

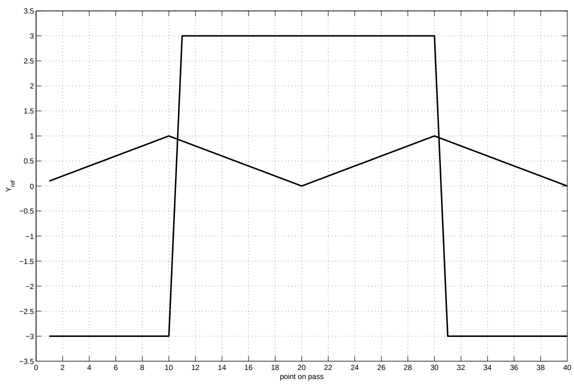
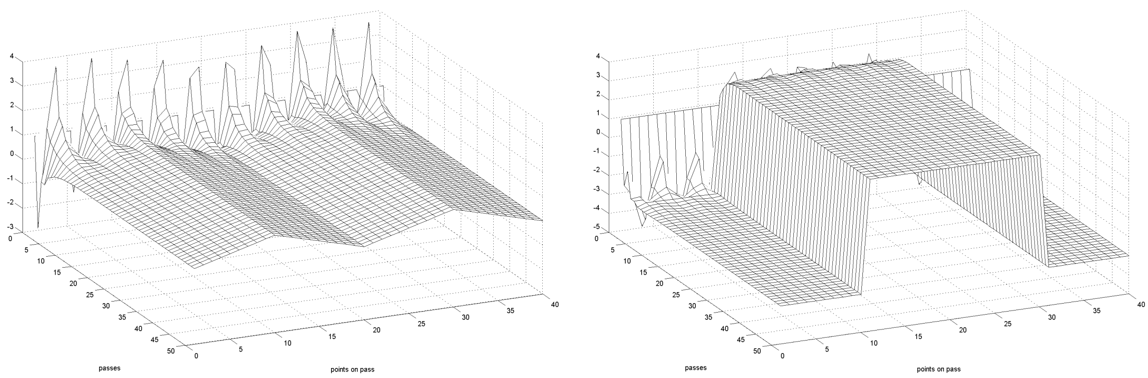


Figure 5.16. The reference signals



Channel 1 Channel 2

Figure 5.17. Pass profiles for the reference signal shown in Figure 5.14

If this condition holds then the matrices in the control law are given by

$$\begin{bmatrix} \widetilde{K}_{y1} & \widetilde{K}_{\chi1} & \widetilde{K}_{y2} & \widetilde{K}_{\chi2} \end{bmatrix} = NX^{-1}.$$

Proof. Note that the above LMI is the output controller design condition for the dynamical process described as the discrete LRP. Hence the proof is the same as that for Theorem 4.18. □

Suppose now that this last result holds. Then it follows immediately that $y_\infty(p) = y_{ref}(p)$, as required. Moreover, using the original variables

$$u_{k+1}(p) = u_\infty(p) + \begin{bmatrix} \widetilde{K}_{y1} & \widetilde{K}_{\chi1} & \widetilde{K}_{y2} & \widetilde{K}_{\chi2} \end{bmatrix} \begin{bmatrix} y_{k+1}(p) - y_{ref}(p) \\ \chi_{k+1}(p) - \chi_\infty(p) \\ y_k(p) - y_{ref}(p) \\ \chi_k(p) - \chi_\infty(p) \end{bmatrix}$$

and also

$$-\widetilde{K}_{y1}y_{ref}(p) - \widetilde{K}_{\chi1}\chi_\infty(p) - \widetilde{K}_{y2}y_{ref}(p) - \widetilde{K}_{\chi2}\chi_\infty(p) + u_\infty(p) = 0.$$

Hence the final form of the control law to be applied to the original process becomes

$$u_{k+1}(p) = \widetilde{K}_{y1}y_{k+1}(p) + \widetilde{K}_{\chi1}\chi_{k+1}(p) + \widetilde{K}_{y2}y_k(p) + \widetilde{K}_{\chi2}\chi_k(p). \tag{5.72}$$

Example 5.9 *Consider the unstable along the pass case of (5.1)-(5.2) given by*

$$A = \begin{bmatrix} 0.48 & -0.45 & -0.11 & -0.07 \\ -0.14 & -0.38 & -0.02 & -0.28 \\ -0.31 & -0.2 & 0.64 & 0 \\ 0 & 0.89 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1.65 & -0.3 & 0.89 & -1.85 & -0.46 \\ 1.96 & 2.0 & -2.88 & -1.47 & 1.54 \\ 0.86 & 0.9 & 0.3 & -2.78 & 0.42 \\ -0.31 & -0.03 & 0.38 & -2.52 & -1.66 \end{bmatrix}, E = \begin{bmatrix} 0.9 \\ -0.54 \\ 0.21 \\ -0.03 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 0.91 & 1.12 & -0.55 \\ 0.07 & -0.15 & 0 \\ 0.64 & 0.8 & 0.72 \\ -0.35 & -1.23 & -1.34 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 0.96 & 0.46 \\ -0.9 & 0 & 0.31 & 0.52 \\ -0.5 & -0.7 & -0.88 & 0.86 \end{bmatrix}, D_0 = \begin{bmatrix} 0.62 & -1.25 & 1.49 \\ -1.67 & -1.33 & 1.64 \\ -0.6 & 0.83 & -0.41 \end{bmatrix},$$

$$D = \begin{bmatrix} 0 & 0 & -2.95 & -2.59 & 1.98 \\ 1.71 & -1.15 & -2.03 & -0.06 & 0 \\ 0 & 0.96 & -2.68 & 0 & 2.68 \end{bmatrix}, F = \begin{bmatrix} 0.78 \\ 0.52 \\ -0.09 \end{bmatrix},$$

over the pass length $\alpha = 200$.

The application of Theorem 5.4 now gives the following control law matrices

$$\begin{bmatrix} \widetilde{K}_{y1} \; \widetilde{K}_{\chi1} \end{bmatrix} = \begin{bmatrix} 4.0771 & -73.2805 & 258.1872 & -50.3334 & -94.4855 & -294.2185 \\ -9.0022 & 333.8740 & -1150.4221 & 212.6543 & 425.1280 & 1308.7428 \\ -7.7820 & -161.6879 & 531.2070 & -82.6427 & -202.0817 & -602.8251 \\ 0.9191 & 178.8760 & -603.4201 & 104.2615 & 225.5047 & 685.4912 \\ -11.9540 & -348.1242 & 1154.4250 & -185.6538 & -435.9741 & -1311.0963 \end{bmatrix},$$

$$\begin{bmatrix} \widetilde{K}_{y2} \; \widetilde{K}_{\chi2} \end{bmatrix} = \begin{bmatrix} 0.1532 & -1.4666 & -0.9546 & 45.0450 & 70.3964 & 284.7620 \\ -1.2662 & 3.5121 & 0.6782 & -190.0204 & -316.6087 & -1266.4867 \\ 0.7462 & -1.7986 & 0.2572 & 73.9629 & 150.6724 & 582.5045 \\ -0.7509 & 1.7314 & -0.3246 & -93.1951 & -167.9548 & -663.0818 \\ 1.5462 & -2.9682 & 0.2081 & 166.0077 & 324.9438 & 1267.0353 \end{bmatrix}.$$

Due to the space limitations in the dissertation, the figures illustrating the outputs of the considered system under control are skipped.

5.4.5 Differential LRPs – stability along the pass – output control

Analogously, as it had a place for discrete LRP, it is possible as well to link the concept of output controller design with PI control scheme for differential LRPs. Note that for the incremental model of (5.62)-(5.63) it is also possible to define the output control law. Remind that in the incremental model of (5.62)-(5.63) the disturbances have been rejected. Hence define the output control law for that incremental model as

$$\begin{aligned}\hat{u}_{k+1}(t) &= \widetilde{K}_1\hat{z}_{k+1}(t)+\widetilde{K}_2\hat{z}_k(t) \\ &= \widetilde{K}_{11}\hat{y}_{k+1}(t)+\widetilde{K}_{12}\hat{\chi}_k(t)+\widetilde{K}_{21}\hat{y}_k(t)+\widetilde{K}_{22}\hat{\chi}_k(t) \\ &= \left[\begin{array}{cccc}\widetilde{K}_{11} & \widetilde{K}_{12} & \widetilde{K}_{21} & \widetilde{K}_{22}\end{array}\right]\left[\begin{array}{c}\hat{y}_{k+1}(t)\\ \hat{\chi}_k(t)\\ \hat{y}_k(t)\\ \hat{\chi}_k(t)\end{array}\right].\end{aligned} \tag{5.73}$$

Note that (5.73) can be treated as a special case of (4.71) since $z_k(t)$ is now the extended pass profile vector. Hence (5.73) can be presented as

$$u_{k+1}(t)=(I-\widetilde{K}_1\hat{D})^{-1}\widetilde{K}_1\hat{C}\hat{x}_{k+1}(t)+(I-\widetilde{K}_1\hat{D})^{-1}[\widetilde{K}_2+\widetilde{K}_1\hat{D}_0]\hat{z}_k(t) \tag{5.74}$$

and it is straightforward to see that (5.74) can be treated as a particular case of (5.73) with

$$\begin{aligned}K_x &= (I-\widetilde{K}_1\hat{D})^{-1}\widetilde{K}_1\hat{C}, \\ K_z &= (I-\widetilde{K}_1\hat{D})^{-1}(\widetilde{K}_2.+\widetilde{K}_1\hat{D}_0)\end{aligned} \tag{5.75}$$

This route may again, however, encounter serious numerical difficulties (arising from the fact that (5.75) is a set of matrix nonlinear algebraic equations) and hence proceed by rewriting these last equations to obtain

$$\begin{aligned}(I-\widetilde{K}_1\hat{D})K_x &= \widetilde{K}_1\hat{C}, \\ (I-\widetilde{K}_1 D)K_z &= \widetilde{K}_2+\widetilde{K}_1\hat{D}_0\end{aligned}$$

and assume that $K_x = L_x\hat{C}$. Now, it follows immediately that

$$\begin{aligned}\widetilde{K}_1 &= L_x(I+\hat{D}L_x)^{-1}, \\ \widetilde{K}_2 &= [I-L_x(I+\hat{D}L_x)^{-1}\hat{D}]K_z - L_x(I+\hat{D}L_x)^{-1}\hat{D}_0\end{aligned} \tag{5.76}$$

for any L_x such that $I+\hat{D}L_x$ is nonsingular, and the following result is obtained. It shows how to design this control law to ensure that (5.62)-(5.63) is stable along the pass under the chosen output control.

Theorem 5.5 *Suppose that the model of (5.62)-(5.63) is subject to a control law of the form of (5.73). Then the resulting closed loop process is stable along the pass if there exist matrices $\hat{Y}>0$, $\hat{Z}>0$, $\hat{X}>0$, $\hat{M}$ and $\hat{N}$ such that the following LMI holds*

$$\left[\begin{array}{ccc}\hat{Y}A^T+A\hat{Y}+\hat{C}^T\hat{N}^TB^T+B\hat{N}\hat{C} & \hat{B}_0\hat{Z}+B\hat{M} & \hat{Y}\hat{C}^T+\hat{C}^T\hat{N}^T\hat{D}^T \\ \hat{Z}\hat{B}_0^T+\hat{M}^TB^T & -\hat{Z} & \hat{Z}\hat{D}_0^T+\hat{M}^T\hat{D}^T \\ \hat{C}\hat{Y}+\hat{D}\hat{N}\hat{C} & \hat{D}_0\hat{Z}+\hat{D}\hat{M} & -\hat{Z}\end{array}\right] < 0, \tag{5.77}$$

$$\hat{X}\hat{C} = \hat{C}\hat{Y}. \tag{5.78}$$

If this condition holds, the control law matrices L_x and K_z are computed by

$$L_x = \hat{N}\hat{X}^{-1}, \quad K_z = \hat{M}\hat{Z}^{-1}. \tag{5.79}$$

Proof. Simply note that (5.62)-(5.63) is of the form (2.14)-(2.15) and hence Theorem 5.3 can be applied to the closed loop process state-space model. □

When the controller matrices computed using (5.79) are available, then it is necessary to apply (5.76) to compute the required output controllers.

To present how (5.73) can be actually employed, first note that

$$\begin{aligned} \hat{u}_{k+1}(t) &= u_{k+1}(t) - u_\infty(t) \\ &= \widetilde{K}_1 \begin{bmatrix} y_{k+1}(t) - y_{ref}(t) \\ \chi_{k+1}(t) - \chi_\infty(t) \end{bmatrix} + \widetilde{K}_2 \begin{bmatrix} y_k(t) - y_{ref}(t) \\ \chi_k(t) - \chi_\infty(t) \end{bmatrix} \end{aligned} \tag{5.80}$$

or, using the original variables,

$$\begin{aligned} u_{k+1}(t) &= \widetilde{K}_{11}\Big(y_{k+1}(t) - y_{ref}(t)\Big) + \widetilde{K}_{12}\Big(\chi_{k+1}(t) - \chi_\infty(t)\Big) \\ &\quad + \widetilde{K}_{21}\Big(y_k(t) - y_{ref}(t)\Big) + \widetilde{K}_{22}\Big(\chi_k(t) - \chi_\infty(t)\Big) + u_\infty(t). \end{aligned} \tag{5.81}$$

This control law can also be applied to the process in non-incremental form, i.e. as

$$u_{k+1}(t) = \widetilde{K}_{11}y_{k+1}(t) + \widetilde{K}_{12}\chi_{k+1}(t) + \widetilde{K}_{21}y_k(t) + \widetilde{K}_{22}\chi_k(t) \tag{5.82}$$

or

$$u_{k+1}(t) = (\widetilde{K}_{11} + \widetilde{K}_{12})y_{k+1}(t) + \widetilde{K}_{21}y_k(t) + (\widetilde{K}_{22} + \widetilde{K}_{12})\chi_k(t) - \widetilde{K}_{12}y_{ref}(t). \tag{5.83}$$

Then from (5.80) it is straightforward to see that

$$-(\widetilde{K}_{11} + \widetilde{K}_{21})y_{ref}(t) - (\widetilde{K}_{12} + \widetilde{K}_{22})\chi_\infty(t) + u_\infty(t) = 0. \tag{5.84}$$

Again, on any pass it is not required to know information which is generated on future passes, i.e. $\chi_\infty(t)$ and $u_\infty(t)$, which considerably simplifies the effort required to construct the control law output to be applied to the process since there is no need to pre-compute these two terms.

Example 5.10 *Consider again the model of unstable (5.3)-(5.4), given in Example 4.5 (or Example 5.8) with the pass length $\alpha = 50$. The application of Theorem 5.5 provide the following matrices*

$$\hat{Y} = \begin{bmatrix} 2235.0880 & 575.9481 & -1439.4247 & 1807.0567 \\ 575.9481 & 7537.5547 & -1617.3023 & 4197.4703 \\ -1439.4247 & -1617.3023 & 2396.1731 & -2207.6201 \\ 1807.0567 & 4197.4703 & -2207.6201 & 5138.4367 \end{bmatrix},$$

$$\hat{Z} = \begin{bmatrix} 6085.1448 & -1221.8324 & -201.6942 & 697.9010 \\ -1221.8324 & 9451.2316 & 354.8953 & -202.0635 \\ -201.6942 & 354.8953 & 8571.7929 & -332.5515 \\ 697.9010 & -202.0635 & -332.5515 & 9048.3441 \end{bmatrix},$$

$$\hat{X} = \begin{bmatrix} 2481.4879 & 0 & 0 & 4011.1455 \\ 0 & 10139.1295 & 4011.1455 & 0 \\ 0 & 4011.1455 & 2481.4879 & 0 \\ 4011.1455 & 0 & 0 & 10139.1295 \end{bmatrix},$$

$$\hat{N} = \begin{bmatrix} 0 & 0 & -1153.3678 & -9076.1548 \\ 767.5183 & 6531.1174 & 0 & 0 \\ 0 & 0 & 530.9927 & 1126.6878 \end{bmatrix}$$

$$\hat{M} = \begin{bmatrix} 292.3406 & -2264.1748 & 163.4142 & -337.4437 \\ -963.2252 & 1601.6416 & 225.3332 & -317.5850 \\ -638.6351 & -199.1936 & 276.9223 & 327.3758 \end{bmatrix},$$

The state/output controllers computed according to (5.79) and assumption $K_x = L_x\hat{C}$ *become*

$$L_x = \begin{bmatrix} 4.0135 & 0.5100 & -1.2892 & -2.4829 \\ 0.8579 & 1.7867 & -2.8881 & -0.3394 \\ -0.4982 & -0.2348 & 0.5935 & 0.3082 \end{bmatrix}$$

and

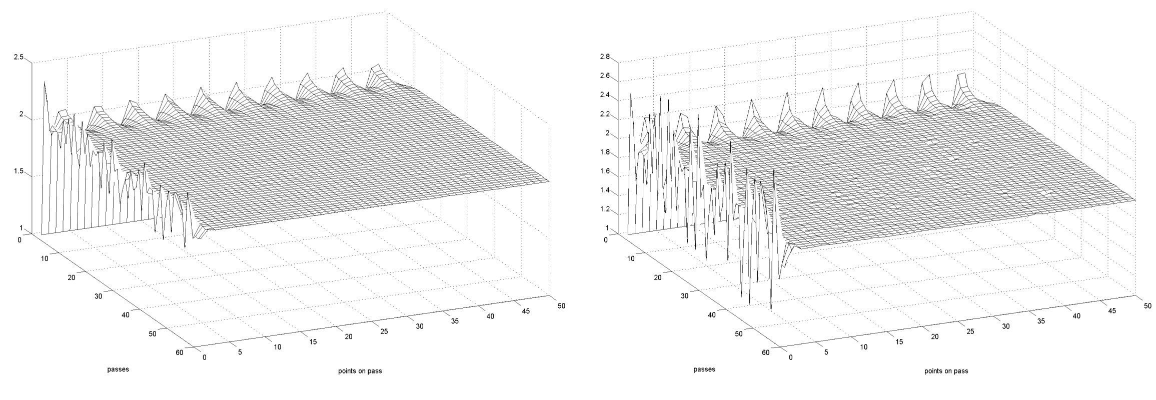
$$K_x = \begin{bmatrix} -1.3621 & 0.0940 & -1.3340 & 0.9113 \\ 1.0151 & -0.0563 & 0.9850 & -0.6654 \\ -0.0477 & -0.0822 & 0.0103 & -0.0536 \end{bmatrix}, \; K_z = \begin{bmatrix} 0.0054 & -0.2408 & 0.0275 & -0.0421 \\ -0.1247 & 0.1523 & 0.0162 & -0.0215 \\ -0.1164 & -0.0364 & 0.0328 & 0.0456 \end{bmatrix}.$$

Hence the output PI controllers applicable in (5.82), computed using (5.76) become

$$\widetilde{K}_{11} = \begin{bmatrix} 1727.1714 & 958.3936 \\ -1221.8575 & -678.0020 \\ -321.5754 & -178.1257 \end{bmatrix}, \; \widetilde{K}_{12} = \begin{bmatrix} -1662.4047 & -979.1791 \\ 1175.8137 & 692.6759 \\ 310.0464 & 182.4651 \end{bmatrix},$$

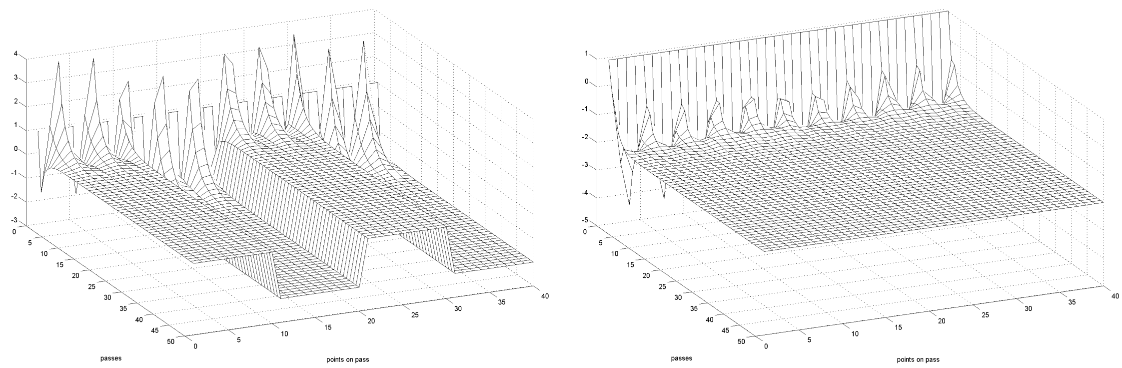
$$\widetilde{K}_{21} = \begin{bmatrix} 2.2496 & 0.2697 \\ -1.7322 & -0.2185 \\ -0.4408 & -0.0783 \end{bmatrix}, \; \widetilde{K}_{22} = \begin{bmatrix} 1688.4483 & 965.6670 \\ -1194.2895 & -683.1710 \\ -314.6651 & -179.7127 \end{bmatrix}.$$

The disturbances, the boundary conditions and the reference signal have been assumed to be the same as in Example 5.8. Figure 5.18 shows the response of the resulting closed loop process under the output control. This confirms that the design objectives have been satisfied.



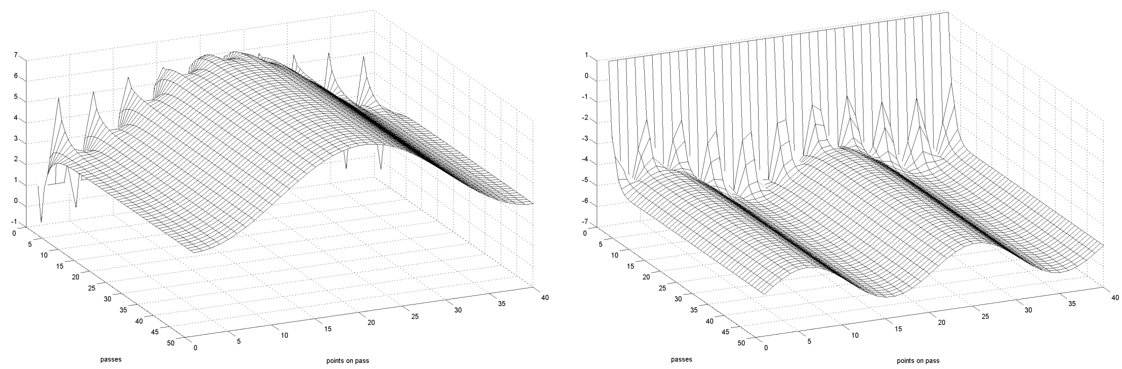
Channel 1   Channel 2

Figure 5.18. The closed loop process pass profile dynamics for the reference signals $[2\ 2]^T$



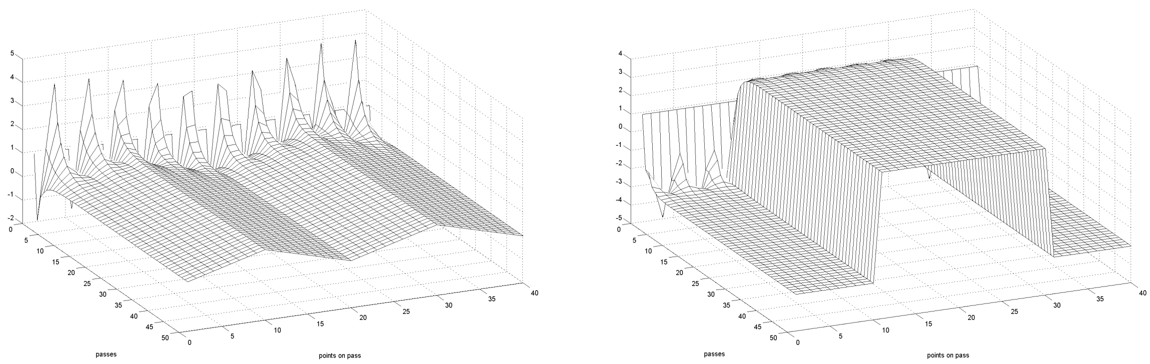
Channel 1 Channel 2

Figure 5.19. Pass profiles for the reference signal shown in Figure 5.12



Channel 1 Channel 2

Figure 5.20. Pass profiles for the reference signal shown in Figure 5.14



Channel 1 Channel 2

Figure 5.21. Pass profiles for the reference signal shown in Figure 5.16

To confirm the usefulness of the method, the simulations for the same reference signals as in Example 5.8 have been made. The respective closed loop process pass profile dynamics are shown in Figures 5.19, 5.20 and 5.21 respectively. It is easy to see that in this case no visible differences in the process dynamics when using the state or the output controller.

5.5 Summary

In this chapter the control for performance of LRPs have been considered. There have been presented four control schemes governing the stability (asymptotic either along the pass) and requested reference signal y_{ref} under assumed control. The application of the output-based controllers is also possible for the considered control schemes. The additional result is the rejection of the disturbances. However, the class of disturbances considered here appear to be narrow (do not change from pass to pass) but only this assumption ensures the ability to totally reject the disturbances. One practically relevant alternative is to seek to attenuate the effects of disturbances to a prescribed degree using, for example, H_∞, H_2, or mixed H_2/H_∞, techniques. In which context some significant first results in this direction can be found in [151].

Conclusions and future works

Analysis and synthesis of nD systems (and Linear Repetitive Processes in particular) is not a trivial work. Solving those problems requires the application of the special conditions (from the theoretical standpoint) and has to be supported by using the powerful computational machines and efficient software packages. The originally existing stability conditions (analysis problem), based on definitions or the 2D transfer function, have the limited applicability and do not provide the natural extension to the controller design (synthesis). They also do not deal with additional aspects e.g. assuring the stability margins. Hence in this dissertation, the LMI conditions for analysis and synthesis of LRPs are presented and used as a basis for the possible extensions. It is to note that the solutions to the stated problems are obtained thanks to the application of the approaches which originally have their sources deep in the Computer Science area. Also, LRPs alone have strong links to this world, as they can be used to model iterative computational processes, and their stability represents the iterative process convergence.

In this dissertation two basic types of stability for the LRPs are considered. Those include asymptotic stability and stability along the pass. Both of those stabilities can cause serious problems when trying to solve related to them tasks of analysis and/or synthesis. The obstacles which appear include dealing with matrices of the possible huge dimensions (asymptotic stability) or checking the condition value for the infinite number of combinations of two complex numbers (stability along the pass). The application of LMIs provides the ability to solve those problems in polynomial to the size of the problem (actually, polynomial to the number of optimization/decision variables) time.

Regarding stability along the pass, it is to remain that existing analysis/synthesis conditions have been treated as being the $\mathcal{NP}$-hard, however it is possible to formulate them to the approximated $\mathcal{P}$ problems and the application of the LMI methods allows to solve such problems efficiently from the numerical standpoint. However, considered resulted LMI conditions are only the sufficient, instead of original necessary and sufficient ones. Hence this fact (skipping the necessity of the conditions) can be treated as a "price", which is to be paid for the possibility to solve the considered problem. There have been defined two distinct LMI conditions which essentially base on the possibility to present the considered model of LRP as RM or FM respectively. Obtained results prove that those conditions work well for considered class of problems and, what is more, after the appropriate reformulation of the problem, they can be assumed to approach the necessary and sufficient conditions.

The basic problem, which appears when asymptotic stability is considered, is the dimensionality of the problem. This is related to the fact that in such a case the so-called 1D equivalent model of LRP is investigated. Hence some methods trying to solve that problem have been presented and tested. First of those is the idea of taking the advantage of the computational power, which is governed by the application of the parallel computing techniques. In the dissertation the approach how to reformulate the original LMI problem into, the solvable on the cluster, SDP problem has been presented. There have been run several tests, which prove the applicability of the clusters. Note that performed tests show that the computational complexity for the considered problems can be approximated successfully by the polynomial of the second degree and the increasing the number of nodes in the cluster causes significant acceleration of the computations. The advantages of the tested cluster are: providing the way to solve the highly dimensional problems and the significant acceleration of the computations.

The other developed approaches to handle problems of large (possibly huge) dimensions are the simplification of the original synthesis task (here called the decoupling of the dynamics) or the application of the iterative approach (the development of the so-called successive stabilization algorithm). It should be underlined that the techniques which have their in the Computer Science, have been successfully employed here to solve the regarded analysis/synthesis problems.

Besides the "basic" LMI conditions, ensuring those features (for stability investigation and controller design), several additional aspects regarding the additional dynamical properties of the considered system have been considered as well. For asymptotic stability that includes 1D model matching. For stability along the pass, the improvements of the basis conditions include: computation of 2D stability margins, synthesis to prescribed stability margins and 2D model matching. There has been developed the controller design scheme basing upon the information from the pass profile (output) vectors. The topics enlisted above are the theoretical results of this dissertation. Nevertheless, the numerical examples have been provided for all of the enlisted topics to highlight them. The series of numerical tests have been performed to test and compare presented methods.

Developing the appropriate LMI conditions for analysis and synthesis provides the strong theoretical result, its applicability can be seriously limited in cases when dealing with large-dimensioned systems. This situation takes a place, when the 2D approaches are used for the considered LRP with relatively large numbers of states, inputs and outputs or in particular, when the 1D equivalent model of LRP of the huge dimensionality is considered. In those cases two possible approaches are feasible to use: first - try to simplify the structure of the considered problem and then solve it numerically (those are addresses in Sections regarding the decoupling or successive stabilization); and second - exploit the abilities of the parallel computing, i.e. use the powerful computational computer cluster with the appropriate solvers installed to solve the large problems. It is of course purposeful to try to joint the two above ways, i.e. first simplify the problem and then, solve it on the cluster. As a practical (implementational) results of this dissertation the methods how to treat, programm and finally solve the LMI conditions using the parallel computing, is proposed. The presented results prove the applicability of parallel computing techniques in analysis and synthesis of LRPs.

Note that in context of physical systems (which are modeled as LRPs) checking/assuring the stability is only the preliminary step. On average a physical system has to do some work and its result can be treated as obtaining the required output value. This can be seen especially well when considering the metal rolling process. Its aim is to reduce the thickness of the rolled bar to required value. Due to that motivation, as a step further to stability/stabilization, the control schemes for driving the controlled system to the required output value have been considered. Hence the results regarding the stability of the system are extended. Another topics considered simultaneously with driving the system to the reference signal is the rejection of the external disturbances. In this dissertation four control schemes are presented, i.e. direct, indirect, feedforward feedback and proportional integral. To present the operation of those schemes the simulation examples have been provided.

Finally, again it is to emphasize that LRPs and LMI methods have close links to computer science as belonging to some extent to the algorithmic area. This aspect of the dissertation is enforced by analytical and numerical tools developed or extended here to simplify the original complex problem or, at last, to make it possible to solve the problem at all.

According to the practical and theoretical results described above (and in the dissertation) it can be concluded that the stated thesis has been proved.

Future works

The results presented in this dissertation pass over some aspects which appear in the analysis and synthesis tasks of LRPs. First, very important are the uncertainty topics related to the fact that LRPs model the physical systems. Hence the appearance of the model uncertainties is the natural consequence of the modeling. Those aspects are related close to the robust control and are still under development. Some preliminary results regarding the uncertain LRPs analysis can be found e.g. in [48, 152, 153, 20] and it is to note that the author of this dissertation is also involved into hat work.

Another possible direction of research on LRPs appears in the possible applications area. As aforementioned, ILC schemes are the iterative procedures and there exist strong theoretical links, which connect the stability of considered LRPs and the convergence of the modeled ILC schemes.

Recently, in [154] the problem of the compensator design for self servo-writing in disk drives has been presented as one, which can be considered and solved as a LRP synthesis problem. The other possible applications come from the fact that spatially interconnected systems (refer to [32]) after the appropriate formulations can be modeled as LRPs.

The research work on those topics is still under development and the possible results will be reported in due course.

Appendix A - description of the selected LMI/SDP solvers

A.1 Matlab - LMI Control Toolbox

Among many applicable scripts and functions it provides the demonstration script called *lmidem*. It provides the summary of the main properties due to defining, solving and interpreting the results of the solved LMIs.

To facilitate defining the variables and constrains, MATLAB LMI CONTROL TOOLBOX provides the parser script called *lmiedit*. It allows to define variables together with its dimensions and its structures and formulate any constraint required in the form similar to the standard MATLAB expressions. As a result of calling *lmiedit* one gets a block of MATLAB code applicable in the further editing. Hence it can be pasted directly into any MATLAB script or the function. Nevertheless, it is worth to mention that *lmiedit* has a limitation regarding the defining the variables. Namely, when it is required to define the variable matrix of a specific structure (e.g. diagonal or sparse), it should be done "by hand".

It is worth mentioning here that during defining LMI with *lmiedit*, it is necessary to define only the lower and the diagonal part of the considered LMI. The rest of blocks can be replaced with $(*)$. This nice property comes from the symmetry of LMI – since it is symmetric, it is enough to define only the "half" of the constraint.

A.1.1 Feasibility problem

Finding any feasible solution x to the LMI system:

$$F(x) < 0 \tag{5.85}$$

is called the *feasibility problem*. Probably, the best example for presenting this problem is the dynamical system stability investigation. For the continuous 1D system of

$$\dot{x}(t) = Ax(t),$$

the LMI regarding its stability takes the form of

$$A^T P + PA < 0, \;\; P = P^T > 0.$$

For the 1D discrete system defined as

$$x_{k+1} = Ax_k, \tag{5.86}$$

the LMI regarding the stability states that, the considered system is stable if and only if there exists $P = P^T > 0$ such that the following LMI holds

$$A^T PA - P < 0. \tag{5.87}$$

Solving of the feasibility problem presented below for the task of the 1D discrete system (5.86) stability investigation. Note that the stability condition in term of LMI is provided by (5.87). The task is now to find out if for a given 1D system matrix A there exists a symmetric matrix $P > 0$, satisfying (5.87).

Example 3.2 is a simple feasibility problem. From the control standpoint the feasibility problem is often used in analysis and synthesis tasks. The following example provides the solution for the stability investigation for 1D discrete system using the feasibility problem.

Example 5.11 *As a example, consider 1D discrete system described by (5.86), where the system matrix A becomes*

$$A = \begin{bmatrix} 0.9 & 0.21 \\ -0.54 & -0.03 \end{bmatrix}.$$

The considered system is stable since the eigenvalues of A become 0.7557, 0.1143*. Application of the above specified procedure provide the following* $tmin = -0.333414$ *and*

$$P = \begin{bmatrix} 1.8108 & 0.4332 \\ 0.4332 & 1.4111 \end{bmatrix}.$$

It means that the feasibility has been proved. What is more, P is symmetric and positive definite since its eigenvalues become 1.1338, 2.088*. The LMI* $A^T PA - P$ *is negative definite since its eigenvalues become* $-0.3370,\ -1.3521$.

Remark 5.5 *It is possible to specify as many variables and LMI constrains of dimensions that can differ due to requirements, but it has to be satisfied that the integrity of the conditions hold. This means that since the multiplication of matrices has a place, all variables has to be of appropriate dimensions and all blocks in defined LMIs together has to built symmetric constrains.*

The process of programming the LMI starts with specifying the matrix variables which are to be computed by the solver. The following lines show how to "tell" the solver what kind of variables are used.

```
setlmis([]);
P=lmivar(1,[n,1]);
```

First line of the above code starts the process of specifying the LMI. Second line stands for that matrix P is symmetric (first input parameter of *lmivar* is equal 1) and of dimensions $n \times n$ (second input parameter of *lmivar* is $[n\ 1]$ - since the matrix is symmetric, i.e. quadratic, only first entry of the vector matters). Standard options for the structures are available: symmetric (value 1), rectangular unstructured (2) and other (3). Third option is used to define the matrix variable of the any required structure e.g. block diagonal with non-zero blocks on the diagonal only. Although this option is available in *lmiedit*, matrices of this kind practically must be defined "by hand".

Next step is the specification of the LMI constraint(s). It is done using the *lmiterm* function.

```
lmiterm([1 1 1 P],A',A);
lmiterm([1 1 1 P],1,-1,);
lmiterm([-2 1 1 P],1,1);
discrete-1D-stability=getlmis;
```

First input parameter of *lmiterm* is the 4-entry vector which entries denotes: 1st - the number (absolute value) of the current LMI being defined and the side (the sign of 1st parameter) of $<$ which is being defined. So, the construction `lmiterm([1 1 1 P],1,1);` denotes $P < 0$ and, on the other hand, `lmiterm([-1 1 1 P],1,1);` denotes $0 < P$ (i.e. $P > 0$). 2nd and 3rd entries of that vector denote the location in the LMI, 4th - the matrix variable which is to be computed and if it is transposed (in such a case $-$ sign is put in the front of the matrix variable). Second and third parameters of *lmiterm* become the multipliers that are standing beside the matrix variable in the defined location. Second denotes the left multiplier and third - right multiplier, respectively. There can be optional 4th parameter, which is the flag $'s'$ and it is used to define the symmetry of the considered block itself.

First line of the above code denotes $A^T PA$ and the second one stand for $-P$. Note that since this two lines treads the same location they are combined together as $A^T PA - P$. Third line of the above code stands for P. So the above code stands for the following stability condition $A^T PA - P < 0$, $P > 0$.

Due to the symmetry of defined LMI it is necessary to define only the upper (or lower) and diagonal part of considered LMI. The rest of blocks can be skipped.

The last line of the above code ends specifying the LMIs. The defined above variable(s) and LMI constrains now can be accessed by reference to the variable `discrete-1D-stability`

Next line calls the feasibility problem solver e.g. function *feasp*. It is straightforward to see that *feasp* is called with set of defined LMIs. There are also some optional parameters related to the accuracy, maximum number of iterations, the stop of the whole procedure condition and tracing of the execution on the screen, which can be added during calling this function.

```
[tmin,Pfeasp]=feasp(discrete-1D-stability);
```

The output values returned after finishing that function are *tmin* which points out if the solved problem was feasible (*tmin* negative) or not (*tmin* positive). In some cases, there arises the situation when *tmin* is positive but very small (rank of 10^{-6} or similar). This means that there appeared some apparent problem during the execution and the solver could not certify the feasibility but, on the other hand, the problem "seemed to be possible feasible". This case can be related to finding such a matrix variables which ensure that set of the eigenvalues of the considered LMI are not strictly negative but they can be almost zero but still positive.

Second of returned variables ($Pfeasp$) combines in its structure all of matrix variables which were sought during the execution of the *feasp* function. Of course it is reasonable to use it only in the case when *tmin* < 0 (feasible case). To extract the original matrix variables which were sought from that variable it is necessary to call the function *dec2mat*.

```
fP=dec2mat(discrete-1D-stability,Pfeasp,P);
```

After this line variable `fP` is the sought Lyapunov matrix. Note that here exist two variables related to the Lyapunov matrix i.e. P and fP. First is used to define the LMI, second is used to store the computed matrix P. Of course it is proper to replace the above line with the following

```
P=dec2mat(discrete-1D-stability,Pfeasp,P);
```

A.1.2 Minimization of the linear function with LMI constraints

Minimizing a convex objective under LMI constrains is the second kind of defined problems solvable by MATLAB Control Toolbox. It is also a convex problem. This is called *linear objective minimization problem* and is defined as follows

$$\min c^T x \text{ subject to } F(x) < 0. \tag{5.88}$$

To provide a highlight of that problem, consider the following optimization problem ([44])

$$\begin{aligned} \min \;\; & \mathrm{trace}(X), \\ \text{subject to } & A^T X + XA + XBB^T X + Q < 0, \\ & X > 0, \end{aligned}$$

where A, B and $Q = Q^T$ are given. For this problem, it can be shown that the minimizer X is simply the stabilizing solution of the algebraic Riccati equation.

It is clear that this problem (the constraint) is not in the form of LMI. Hence apply the Schur complement (Lemma 3.1) to obtain the following problem where constraints are given in LMI form

$$\begin{aligned} \min \;\; & \mathrm{trace}(X), \\ \text{subject to } & \begin{bmatrix} A^T X + XA + Q & XB \\ B^T X & -I \end{bmatrix} < 0, \\ & X > 0, \end{aligned}$$

Note that trace(X) is linear function on X and hence the *mincx* solver can be used to provide the solution.

To programme that task, using MATLAB notation use the following code

```
 setlmis([])
X = lmivar(1,[n 1]) % variable X, full symmetric size nXn
lmiterm([1 1 1 X],1,A,'s');
lmiterm([1 1 1 0],Q);
lmiterm([1 2 2 0],1);
lmiterm([1 2 1 X],B',1);
LMIs = getlmis;
```

Note that specifying of the variables and LMI constrains of the above is done in the same manner as in the previous case. Now, there arises the problem of defining the cost vector c. The assumed structure of X can be achieved in two ways, i.e. when the function *lmivar* has been called - instead of

```
 X = lmivar(1,[n 1]) % variable X, full symmetric size nXn
```

call it as

```
 [X,nn,Xstruct] = lmivar(1,[n 1]) % variable X, full symmetric size nXn
```

and then in variable $Xstruct$ the assumed symmetric structure of X is stored or use function *defcx* afterwards. Since the trace of X is to be minimized, it is easy now to define which entries of X are to be summed in the minimization process. Note that the size of c becomes the total number of the decision variables - here $\frac{n(n+1)}{2}$ and in this case is stored in nn (on general the total number of variables can be obtained calling *decnbr*). Hence when considering the structure of X it is immediate to conclude that the trace of X is the sum of elements numbered $1, 3, 6, \ldots, \frac{n(n+1)}{2}$ and the entries of c labeled in this manner treat the elements on the diagonal of X. It is straightforward to see that the cost vector becomes $c = [1, 0, 1, 0, 0, 1, 0, 0, 0, 1, \ldots, 1]$ and can be defined as

```
 i=1:n;
c=zeros(nn,1);
c(i.*(i+1)./2)=1;
```

Finally, it is necessary to call the *mincx* function as

```
 [copt,fXopt]=mincx(LMIs,c);
```

After the execution, the minimal value of the target function i.e. the minimal trace of X is given in *copt*. Again, as it had a place in the feasibility problem, matrix X has to be extracted from $Xopt$ using the *dec2mat* function as (to distinguish it form the definition of the variable X call it fX)

```
 fX=dec2mat(LMIs,Xopt,X);
```

A.1.3 GEVP

Third kind of problem solvable by this toolbox is called *generalized eigenvalue minimization problem* (GEVP). It is defined as

$$\min \lambda \text{ subject to } \begin{cases} F(x) < \lambda\, G(x), \\ G(x) > 0, \\ H(x) < 0. \end{cases} \tag{5.89}$$

Its name comes from the fact that λ is related to the largest generalized eigenvalue of the pencil $\big(F(x), G(x)\big)$ (it is the largest root of the equation $\det[\lambda G(x) - F(x)] = 0$ - see e.g. [155] for details).

To see the applicability of this kind of problem consider the task of pole placement for 1D discrete system with LMI. This problem has been formulated as $\mathcal{D}$-stability and considered in [156, 45, 143].

The task can be considered as providing such a control sequence in the following form

$$u_k = Kx_k \tag{5.90}$$

that the poles of the considered system of (5.86)

$$x_{k+1} = Ax_k + Bu_k$$

in the closed loop configuration lay inside the specified region on the complex plane. The standard stability condition for the discrete system requires $\mathcal{D}$ to be the interior of the unit disc. In this specific case it is required that the closed loop system poles lay inside the ellipse with the center of $(g, 0)$, vertical radius b and horizontal radius a.

This task can be formulated in term of GEVP as follows. Determine the control law matrix K applicable in (5.90) that minimizes the vertical radius of the ellipse $b > 0$ for the given horizontal radius a and the center of located in $(g, 0)$ on the complex plane. It can be formalized as finding matrices: symmetric $Y > 0$ and N of appropriate dimensions such that the following generalized eigenvalue problem has a solution

$$
\begin{aligned}
&\min b,\\
&\text{subject to}\\
&\begin{bmatrix} 0 & (*) \\ aYA^T + aN^TB^T - aAY - aBN & 0 \end{bmatrix} < b \begin{bmatrix} 2aY & (*) \\ 2gY - YA^T - N^TB^T - AY - BN & 2aY \end{bmatrix},\\
&\begin{bmatrix} 2aY & (*) \\ 2gY - YA^T - N^TB^T - AY - BN & 2aY \end{bmatrix} > 0.
\end{aligned}
\tag{5.91}
$$

For this case the controller matrix can be computed as

$$K = NY^{-1}.$$

The detailed description of this problem can be found in [45].

The process of programming the above problem is done in the same manner as for the previous cases. The declaration of matrix variables and specifying the LMIs are done with the following code

```
 setlmis([]);
Y=lmivar(1,[2*n,1]);
N=lmivar(2,[r,2*n]);
lmiterm([-1 1 1 Y],1,1);    % X>0
lmiterm([-2 1 1 Y],2*a,1);  % this LMI denotes that G(x) >0
lmiterm([-2 2 1 Y],2*g,1);  % this LMI denotes that G(x) >0
lmiterm([-2 2 1 Y],-1,A');
lmiterm([-2 2 1 -N],-1,B');
lmiterm([-2 2 1 Y],-A,1);
lmiterm([-2 2 1 N],-B,1);
lmiterm([-2 2 2 Y],2*a,1);
lmiterm([-3 1 1 Y],2*a,1);  % this LMI denotes b*G(x)>F(x)
lmiterm([-3 2 1 Y],2*g,1);
lmiterm([-3 2 1 Y],-1,A');  % minus means right hand side of this inequality
lmiterm([-3 2 1 -N],-1,B'); % thus b*G(x) (without b)
lmiterm([-3 2 1 Y],-A,1);
lmiterm([-3 2 1 N],-B,1);
lmiterm([-3 2 2 Y],2*a,1);
lmiterm([3 2 1 Y],a,A');
lmiterm([3 2 1 -N],a,B');
lmiterm([3 2 1 Y],-a*A,1);
lmiterm([3 2 1 N],-a*B,1);
LMIs=getlmis;
```

It is to note that there is no explicit multiplication between b and $G(x)$ given, however such a factor is given in definition of the problem (5.89). This is due to the fact that still we deal with the LMI solvers and in general the factor $(bG(x))$ can be treated as a BMI one. Nevertheless, thanks to the specific construction problems of that form can be solved by LMI solver. To call the *gevp* solver use the following construction

```
[b,Opt]=gevp(LMIs,1);
```

Note that the number 1 given in the above calling deals with the 'BMI-like' constraints defined as $F(x) < \lambda\ G(x)$. The value of 1 denotes that only one, last constraint in the definition is 'BMI-like' and hence the solver recognizes it properly. That's why the order of the constraints in the above example has been changed it comparison to the definition (5.89) of the problem. It there would appear two (or, say s) constraints involving b (or in general λ) variable, i.e. $F_1(x) < \lambda\ G_1(x)$ $F_2(x) < \lambda\ G_2(x)$ ($F_s(x) < \lambda\ G_s(x)$) then the calling of the *gevp* should read as

```
[b,Opt]=gevp(LMIs,2);
```

Again, after solving the problem, it is necessary to extract the sought variable matrices

```
 Y=dec2mat(LMIs,Opt,Y);
N=dec2mat(LMIs,Opt,N);
```

Now, use the formula $K = NY^{-1}$ to compute the required controller.

A.2 Scilab LMI Optimization Package

Scilab LMI Optimization Package provides two main functions which should be used when there arises the necessity to solve an LMI.

Function *lmitool* provides the user-friendly interface which allows to define LMI problems in the intuitive manner. It has been equipped with very solid help. It is comparable to included in Matlab LMI Control Toolbox *lmiedit*. Nevertheless, it suffers the similar disadvantages as *lmiedit*, i.e. it is applicable only in cases when the constraints and matrix variables to be defined are of the simple structure – it does not allow to define more sophisticated structures in the simple way.

The second important function is *lmisolver*. Indeed, it is the parser function, which prepares the input data provided in the LMI form (constraints, data, matrix variables structures) and finally call the function *semidef*, which is just the SDP solver (implementing the IPM algorithm from [125]). Hence it is the interface to *semidef*. Essentially, the function *lmisolver* works in four steps ([121]):

Initial set-up. The sizes and structures of the initial guess are used to set up the problem and in particular the size of the unknown vector.

Elimination of the equality constraints. Performing the repeated calls to the objective function, the canonical form of the LMI problem with the possible occurrence of the equality constraints is generated. This step uses extensively sparse matrices to speed up the computation and reduce the memory requirements.

Elimination of redundant variables. The equality constraints are eliminated. At this stage, all solutions of the equality constraints are parameterized. This step involves using the sparse LU factorization functions. Once, this process is finished, the original problem is turned into the primal form of SDP given by (3.10).

Optimization. Finally, *lmisolver* calls the *semidef* function and optimization process starts. Itself, it is divided into two stages, i.e. the feasibility stage and optimization stage.

Appendix B - codes of the selected functions

B.1 Matlab

Stability along the pass (discrete) - Theorem 2.3

```
function [t_min,fP,fQ] = is_stable2D(A,B0,C,D0);

[n1,n2]=size(A);
[m1,m2]=size(D0);

A1=zeros(n1+m1,n2+m2);
A2=zeros(n1+m1,n2+m2);
A1(1:n1,1:n2)=A;
A1(1:n1,n2+1:n2+m2)=B0;
A2(n1+1:n1+m1,1:n1)=C;
A2(n1+1:n1+m1,n2+1:n2+m2)=D0;

  setlmis([]);
  P=lmivar(1,[n1+m1,1]);
  Q=lmivar(1,[n1+m1,1]);

  lmiterm([1 1 1 P],A1',A1);                   % LMI #1: A1'*P*A1
  lmiterm([1 1 1 Q],1,1);                      % LMI #1: Q
  lmiterm([1 1 1 P],1,-1);                     % LMI #1: -P
  lmiterm([1 2 1 P],A2',A1);                   % LMI #1: A2'*P*A1
  lmiterm([1 2 2 P],A2',A2);                   % LMI #1: A2'*P*A2
  lmiterm([1 2 2 Q],1,-1);                     % LMI #1: -Q
  lmiterm([2 1 1 P],1,-1);                     % LMI #2: -P
  lmiterm([3 1 1 Q],1,-1);                     % LMI #3: -Q

  is_stable2=getlmis;

[t_min,X_feasp]=feasp(is_stable2);

fP=dec2mat(is_stable2,X_feasp,P);
fQ=dec2mat(is_stable2,X_feasp,Q);
```

2D controller design (discrete) - Theorem 4.9

```
function[t_min,K1,K2]=stabilize_2D(A,B,B0,C,D,D0);
%  [Z-Y   0   Y*A1'+N'*B1' ;
%   0   -Z    Y*A2'+N'*B2' ;
%   A1*Y+B1*N   A2*Y+B2*N     -Y] < 0
%0<Y
%0<Z

[n1,n2]=size(A);
```

```
[m1,m2]=size(D0);
[r1,r2]=size(B);

A1=zeros(n1+m1,n2+m2);
A2=zeros(n1+m1,n2+m2);
B1=zeros(n1+m1,r2);
B2=zeros(n1+m1,r2);
A1(1:n1,1:n2)=A;
A1(1:n1,n2+1:n2+m2)=B0;
A2(n1+1:n1+m1,1:n1)=C;
A2(n1+1:n1+m1,n2+1:n2+m2)=D0;
B1(1:n1,1:r2)=B;
B2(n1+1:n1+m1,1:r2)=D;

  setlmis([]);
  Z=lmivar(1,[n1+m1,1]);
  Y=lmivar(1,[n1+m1,1]);
  N=lmivar(2,[r2,n1+m1]);

  lmiterm([1 1 1 Z],1,1);                        % LMI #1: Z
  lmiterm([1 1 1 Y],1,-1);                       % LMI #1: -Y
  lmiterm([1 2 2 Z],1,-1);                       % LMI #1: -Z
  lmiterm([1 3 1 Y],A1,1);                       % LMI #1: A1*Y
  lmiterm([1 3 1 N],B1,1);                       % LMI #1: B1*N
  lmiterm([1 3 2 Y],A2,1);                       % LMI #1: A2*Y
  lmiterm([1 3 2 N],B2,1);                       % LMI #1: B2*N
  lmiterm([1 3 3 Y],1,-1);                       % LMI #1: -Y
  lmiterm([-2 1 1 Y],1,1);                       % LMI #2: Y
  lmiterm([-3 1 1 Z],1,1);                       % LMI #3: Z

  dis_stabilization=getlmis;

[t_min,X_feasp]=feasp(dis_stabilization, [0 0 0 0 1]);

fY=dec2mat(dis_stabilization,X_feasp,Y) ;
fZ=dec2mat(dis_stabilization,X_feasp,Z);
fN=dec2mat(dis_stabilization,X_feasp,N) ;

Kpom=fN*fY^(-1);
if t_min <0
    K1=Kpom(:,1:n1);
    K2=Kpom(:,n1+1:n1+m1);
   else
      disp('!WARNING! Stabilization uncomplete!');
      K1=zeros(size(B'));
      K2=zeros(size(D'));
 end
```

2D controller design (discrete) - Theorem 4.10

```
function[t_min,K,fN,fP]=stabilize2D2(A,B,B0,C,D,D0);
%  [-P          A_fal*P+B_fal*N;
%   (A_fal*P+B_fal*N)^T   -P] < 0
%0<P

[n1,n2]=size(A);
[m1,m2]=size(D0);
[r1,r2]=size(B);

A_fal=[A B0;C D0];
B_fal=zeros(n1+m1,2*r2);
B_fal(1:n1,1:r2)=B;
B_fal(n1+1:n1+m1,r2+1:2*r2)=D;
```

```
    setlmis([]);
    [P1_pom,n,P11_pom]=lmivar(1,[n1,1]);
    [P2_pom,n,P22_pom]=lmivar(1,[m1,1]);
    [N1_pom,n,N11_pom]=lmivar(2,[r2,n1]);
    [N2_pom,n,N22_pom]=lmivar(2,[r2,m1]);

   P_struct=zeros(n1+m1,n1+m1);
   P_struct(1:n1,1:n1)=P11_pom;
   P_struct(n1+1:m1+n1,n1+1:m1+n1)=P22_pom;
   P=lmivar(3,P_struct);
   N=lmivar(3,[N11_pom N22_pom; N11_pom N22_pom]);

  lmiterm([1 1 1 P],1,-1);                        % LMI #1: -P
  lmiterm([1 2 1 P],1,A_fal');                    % LMI #1: P*A_fal'
  lmiterm([1 2 1 -N],1,B_fal');                   % LMI #1: N'*R'
  lmiterm([1 2 2 P],1,-1);                        % LMI #1: -P
  lmiterm([-2 1 1 P],1,1);                        % LMI #2: P

  dis_stabilization=getlmis;

[t_min,X_feasp]=feasp(dis_stabilization,[0 0 0 0 1]);
fN=dec2mat(dis_stabilization,X_feasp,N) ;
fP=dec2mat(dis_stabilization,X_feasp,P);

Kpom=fN*fP^(-1);
if t_min <0
   K1=Kpom(1:r,1:n);
   K1=Kpom(1:r,n+1:n+m);
   else
      disp('!WARNING! Stabilization uncomplete!');
      K1=0;
      K2=0;
 end
```

B.2 Scilab

1D discrete system controller design

```
function [wporzo, X,K]=stabilize_1D(A,B)

  Mbound = 1e3;
  abstol = 1e-16;
  nu = 10;
  maxiters = 100;
  reltol = 1e-16;
  options=[Mbound,abstol,nu,maxiters,reltol];

///////////DEFINE INITIAL GUESS AND PRELIMINARY CALCULATIONS BELOW
[n,r]=size(B);
X_init=zeros(n,n);
N_init=zeros(r,n);

XLIST0=list(X_init, N_init);
errcatch(9999,"continue") ;
[XLIST,OPT]=lmisolver(XLIST0,stabilize_1D,options);

wporzo=1;
if iserror(9999)==1 then
   wporzo=-1;
end
[X, N]=XLIST(:);
```

```
K=N*X^-1;

////////////////EVALUATION FUNCTION///////////////////////////
function [LME,LMI,OBJ]=stabilize_1D_eval(XLIST)

[X,N]=XLIST(:);

////////////////DEFINE LME, LMI and OBJ BELOW
LME=list(X-X');
LMI=list(X-eye(), [X, -X*A'-N'*B'; -A*X-B*N, X ]-eye() );
OBJ=[];
```

Calling and checking the result in SCILAB

```
--> A=rand(3,3)*3;
--> B=rand(3,2);
-->  [ok, X,K]=stabilize_1D(A,B);
--> max(abs(spec(A+B*K)))
```

2D extended output controller design (discrete) - Theorem 4.20

The stabilization function

```
function [wporzo, Z, Y, X, N]=stab_out_ext(A, B0, C, D0, B, D,opcja)
  gstacksize(10000001)

  Mbound = 1e3;
  abstol = 1e-16;
  nu = 10;
  maxiters = 100;
  reltol = 1e-16;
  options=[Mbound,abstol,nu,maxiters,reltol];

//////////DEFINE INITIAL GUESS AND PRELIMINARY CALCULATIONS BELOW
[n,m]=size(B0);
[n,r]=size(B);

X1_init=zeros(n,n);
X2_init=zeros(m,m);
X3_init=zeros(m,m);
X4_init=zeros(m,m);

Y_init=zeros(2*m+2*n,2*m+2*n);
Z_init=zeros(2*m+2*n,2*m+2*n);

N1_init=zeros(r,m);
N2_init=zeros(r,m);
N3_init=zeros(r,m);
N4_init=zeros(r,m);

errcatch(9999,"continue")

XLIST0=list(Z_init, Y_init, X1_init, X2_init, X3_init, X4_init, N1_init, N2_init, N3_init, N4_init)

[XLIST,OPT]=lmisolver(XLIST0,stab_out_ext_eval_full,options);

wporzo=1;
if iserror(9999)==1 then
   wporzo=0;
end
[Z, Y, X1, X2, X3, X4, N1, N2, N3, N4]=XLIST(:);

X=zeros(3*m+n,3*m+n);
```

```
X(1:n,1:n)=X1;
X(n+1:m+n,n+1:m+n)=X2;
X(m+n+1:2*m+n,m+n+1:2*m+n)=X3;
X(2*m+n+1:3*m+n,2*m+n+1:3*m+n)=X4;

N=[N1,         zeros(r,n), zeros(r,m), zeros(r,m);...
   zeros(r,m), zeros(r,n), zeros(r,m), -N3;...
   zeros(r,m), zeros(r,n), zeros(r,m), -N4;...
   zeros(r,m), zeros(r,n), zeros(r,m), N2];

////////////////EVALUATION FUNCTION///////////////////////////

function [LME,LMI,OBJ]=stab_out_ext_eval_full(XLIST);
[Z, Y, X1, X2, X3, X4, N1, N2, N3, N4]=XLIST(:);
////////////////DEFINE LME, LMI and OBJ BELOW

X=zeros(3*m+n,3*m+n);
X(1:n,1:n)=X1;
X(n+1:m+n,n+1:m+n)=X2;
X(m+n+1:2*m+n,m+n+1:2*m+n)=X3;
X(2*m+n+1:3*m+n,2*m+n+1:3*m+n)=X4;

A1=zeros(2*m+2*n,2*m+2*n);
A1(1:n,1:2*m+2*n)=[A, -eye(A), zeros(n,m), B0];
A2=zeros(2*m+2*n,2*m+2*n);
A2(2*n+m+1:2*m+2*n,:)=[C, zeros(m,n), -eye(D0), D0];

Bfal1=[B,          zeros(n,r), zeros(n,r), B ; ...
       zeros(n,r), B,          zeros(n,r), zeros(n,r);...
       zeros(m,r), D,          zeros(m,r), zeros(m,r);...
       zeros(m,r), zeros(m,r), zeros(m,r), zeros(m,r)];

Bfal2=[zeros(n,r), zeros(n,r), zeros(n,r), zeros(n,r) ; ...
       zeros(n,r), zeros(n,r), B,           zeros(n,r);...
       zeros(m,r), zeros(m,r), D,          zeros(m,r);...
       D,          zeros(m,r), zeros(m,r), D];

N=[N1,         zeros(r,n), zeros(r,m), zeros(r,m);...
   zeros(r,m), zeros(r,n), zeros(r,m), -N3;...
   zeros(r,m), zeros(r,n), zeros(r,m), -N4;...
   zeros(r,m), zeros(r,n), zeros(r,m), N2];

Cfal=[C,       zeros(m,n), zeros(m,m), zeros(m,m);...
      zeros(n,n), eye(n,n), zeros(n,m), zeros(n,m);...
      zeros(m,n), zeros(m,n), eye(m,m), zeros(m,m);...
      zeros(m,n), zeros(m,n), zeros(m,m), eye(m,m)];

LME=list(X-X', Y-Y', Z-Z', X*Cfal-Cfal*Y);
LMI=list(X-eye(), Y-eye(), Z-eye(), [Y-Z, zeros(A1), -Cfal'* N'*Bfal1' - Y*A1';...
                       zeros(A1), Z, -Cfal'* N'*Bfal2' - Y*A2';...
                -A1*Y - Bfal1*N*Cfal, -A2*Y - Bfal2*N*Cfal, Y]-eye());
OBJ=[]
```

Computation of the output controllers for model stored in example.mat file

```
//stabilize_output.sci
//script for the output stabilization
mtlb_load example.mat
getf('stab_out_ext.sci');
[n,m]=size(B0);
[n,r]=size(B);
[w, fZ, fY, fX, fN]=stab_out_ext(A, B0, C, D0, B, D);

if w==1,
```

```
      L=fN*inv(fX);
      L1=L(1:r,1:m);
      K1=L1*C;
      K2=L(3*r+1:4*r,n+2*m+1:n+3*m);
      K3=-L(r+1:2*r,n+2*m+1:n+3*m);
      K4=-L(2*r+1:3*r,n+2*m+1:n+3*m);

      K1fal=L1*inv((eye(m,m)+D*L1));
      K2fal=(eye(r,r)-K1fal*D)*K2 - K1fal*D0;
      K3fal=(eye(r,r)-K1fal*D)*K3;
      K4fal=(eye(r,r)-K1fal*D)*K4;
end;
```

calling in SCILAB

```
--> exec('stabilize_output.sci')
```

B.3 SDPA (SDPARA)

Matlab function which prepares the appropriate SDPA (SDPARA) task file

2D controller design (discrete) - Theorem 4.10

```
  function f_parse_stabilize2D2(A,B0,B,C,D0,D)
%function which prepares the data in the form callable by SDPA(RA)
%2D controller design
%
clc
[n,n]=size(A);
[m,r]=size(D);
AA= [A B0; C D0];
BB=[B zeros(n,r); zeros(m,r) D];
nazwa_dat=['stabilizacja2D_' num2str(n) 'X' num2str(r) 'X' num2str(m) '_W.dat']
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%total number of decision variables Z Y N
nbr_of_var = n*(n+1)/2 + m*(m+1)/2 + r*(m+n) +1
tic
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fid=fopen(nazwa_dat, 'w');
fprintf(fid,'"Example 2D stabilization: mDIM = ');
fprintf(fid,'%i ',nbr_of_var);
fprintf(fid,',nBLOCK = 2');
fprintf(fid,',bLOCKsTRUCT = %i %i"\n', 2*(m+n), m+n);
fprintf(fid,'%i = mDIM \n',nbr_of_var);
fprintf(fid,'2 = nBLOCK \n');
fprintf(fid,'(%i, %i) = bLOCKsTRUCT \n', 2*(n+m), m+n);
fprintf(fid,'\n');
%target vector c
for jj=1:nbr_of_var-1
    fprintf(fid,'%i ',0);
end
fprintf(fid,'%i ',1);
fprintf(fid,'\n\n');
%vertices of the canonic form of LMI
  mac=zeros(2*(n+m));
  for j=1:2*(n+m)
        wiersz=mac(j,:);
        fprintf(fid,'%i ',wiersz(1,1:2*(n+m)));
        fprintf(fid,'\n');
  end
    for j=1:(n+m)
```

```
        wiersz=mac(j,:);
        fprintf(fid,'%i ',wiersz(1,1:(n+m)));
        fprintf(fid,'\n');
        %fprintf(fid,', \n');
  end
    fprintf(fid,'\n');

setlmis([]);
[Q1,nn1,QQ1]=lmivar(1,[n 1]);
[Q2,nn1,QQ2]=lmivar(1,[m 1]);
QQ=[QQ1 zeros(n,m); zeros(m,n) QQ2];
for i=1:nn1
    Qpom=zeros(n+m);
    Qpom(QQ==i)=1;
    mac=[Qpom -Qpom*AA'; -AA*Qpom Qpom];
    for j=1:2*(n+m)
        wiersz=mac(j,:);
        for jj=1:2*(n+m)
            if wiersz(1,jj)==0.0
                fprintf(fid,'%i ',0);
            else
                fprintf(fid,'%2.2f ',wiersz(1,jj));
            end
        end
        fprintf(fid,'\n');
    end
    mac=[];
    mac=Qpom;
    for j=1:(n+m)
        wiersz=mac(j,:);
        for jj=1:(n+m)
            if wiersz(1,jj)==0.0
                fprintf(fid,'%i ',0);
            else
                fprintf(fid,'%2.2f ',wiersz(1,jj));
            end
        end
        fprintf(fid,'\n');
    end
    fprintf(fid,'\n');
end

[N1,nn2,NN1]=lmivar(2,[r n]);
[N2,nn2,NN2]=lmivar(2,[r m]);
NN=[NN1 NN2; NN1 NN2];
for i=nn1+1:nn2
    i
    Npom=zeros(2*r,n+m);
    Npom(NN==i)=1;
    mac=[zeros(n+m) -Npom'*BB'; -BB*Npom zeros(n+m)];
    for j=1:2*(n+m)
        wiersz=mac(j,:);
        for jj=1:2*(n+m)
            if wiersz(1,jj)==0.0
                fprintf(fid,'%i ',0);
            else
                fprintf(fid,'%2.2f ',wiersz(1,jj));
            end
        end
        fprintf(fid,'\n');
    end
    mac=[];
    mac=zeros(n+m);
    for j=1:(n+m)
```

```
        wiersz=mac(j,:);
        for jj=1:(n+m)
            if wiersz(1,jj)==0.0
                fprintf(fid,'%i ',0);
            else
                fprintf(fid,'%2.2f ',wiersz(1,jj));
            end
        end
        fprintf(fid,'\n');
    end
    fprintf(fid,'\n');
end
%last vertex
  mac=eye(2*(n+m));
  for j=1:2*(n+m)
        wiersz=mac(j,:);
        fprintf(fid,'%i ',wiersz(1,1:2*(n+m)));
        fprintf(fid,' \n');
  end
  mac=[];
  mac=eye(n+m);
  for j=1:(n+m)
        wiersz=mac(j,:);
        fprintf(fid,'%i ',wiersz(1,1:(n+m)));
        fprintf(fid,' \n');
  end
  fprintf(fid,'\n');
ex1=getlmis;
fclose(fid);
nazwa_mat=['s2D_' num2str(n) 'X' num2str(r) 'X' num2str(m)   '_W']
eval(['save '  nazwa_mat])
```

2D output controller design (discrete) with the elimination of the equality constraints - Theorem 4.18

```
function f_parse_stabilize2D2(A,B0,B,C,D0,D)

[n,m]=size(B0);
[n,r]=size(B);
nazwa_dat=['stabilizacja2D_out_bezX_' num2str(n) 'X' num2str(r) 'X' num2str(m) '.dat']
A1=[A B0; zeros(m,n+m)];
A2=[zeros(n,n+m); C D0];
B1=[B; zeros(m,r)];
B2=[zeros(n,r); D];
Cfal=[C zeros(m,m); zeros(m,n) eye(m,m)];
[mC,nC]=size(Cfal);
setlmis([]);
[Y,nnY,YY]=lmivar(1,[n+m 1]);
[X,nnX,XX]=lmivar(2,[2*m 2*m]);
[Z,nnZ,ZZ]=lmivar(1,[n+m 1]);
[N,nnN,NN]=lmivar(2,[r 2*m]);

H=zeros(mC*nC,nnN);
for i=1:nnY
    YCpom=zeros(n+m);
    YCpom(YY==i)=1;
    mac=[-Cfal*YCpom];
    for j=1:mC*nC
       macT=mac';
            if macT(j) ~= 0
               H(j,i)=macT(j);
            end
    end
```

```
end

for i=nnY+1:nnX+nnY
    XCpom=zeros(2*m);
    XCpom(XX==i)=1;
    mac=[XCpom*Cfal];
    for j=1:mC*nC
       macT=mac';
            if macT(j) ~= 0
               H(j,i)=macT(j);
            end
    end
end
NullSpace=null(H,'r');
[l_starych,l_nowych]=size(NullSpace);
nbr_of_var=l_nowych+1;
fid=fopen(nazwa_dat, 'w');
fprintf(fid,'"Example of 2D output stabilization: mDIM = ');
fprintf(fid,'%i ',nbr_of_var);
fprintf(fid,',nBLOCK = 4 "\n');

fprintf(fid,'%i = mDIM \n',nbr_of_var);
fprintf(fid,'4 = nBLOCK \n');
fprintf(fid,'%i %i %i 1 = bLOCKsTRUCT ', 3*(n+m), m+n, m+n);
fprintf(fid,'\n \n');
for jj=1:nbr_of_var-1
    fprintf(fid,'%i ',0);
end
fprintf(fid,'%i ',1);
fprintf(fid,' \n');

fprintf(fid,' \n');
  mac=zeros(3*(n+m));
  for j=1:3*(n+m)
        wiersz=mac(j,:);
       fprintf(fid,'%i ',wiersz(1,1:3*(n+m)));
       fprintf(fid,'\n');
  end
  mac=zeros(n+m);
  for j=1:(n+m)
        wiersz=mac(j,:);
        fprintf(fid,'%i ',wiersz(1,1:n+m));
       fprintf(fid,'\n');
  end
  mac=zeros(n+m);
  for j=1:(n+m)
        wiersz=mac(j,:);
       fprintf(fid,'%i ',wiersz(1,1:n+m));
       fprintf(fid,'\n');
  end
  fprintf(fid,'0 \n \n');

for i=1:l_nowych
   wierzcholek1=zeros(3*(m+n));
   wierzcholek2=zeros(m+n);
   wierzcholek3=zeros(m+n);
   wierzcholek4=zeros(2*m);
   for j=1:nnY
      Ypom=zeros(n+m);
        Ypom(YY==j)=1;
      mac1=[Ypom zeros(n+m) -Ypom*A1';...
         zeros(n+m) zeros(n+m) -Ypom*A2' ;...
         -A1*Ypom -A2*Ypom Ypom];
      wierzcholek1=wierzcholek1+NullSpace(j,i)*mac1;
```

```
      mac2=Ypom;
      wierzcholek2=wierzcholek2+NullSpace(j,i)*mac2;
   end
   for j=nnX+1:nnZ
      Zpom=zeros(m+n);
       Zpom(ZZ==j)=1;
      mac1=[-Zpom zeros(n+m) zeros(n+m) ;...
           zeros(n+m) Zpom zeros(n+m);...
           zeros(n+m) zeros(n+m) zeros(n+m)];
      wierzcholek1=wierzcholek1+NullSpace(j,i)*mac1;
      mac3=Zpom;
      wierzcholek3=wierzcholek3+NullSpace(j,i)*mac3;
   end
   for j=nnZ+1:nnN
      Npom=zeros(r,2*m);
        Npom(NN==j)=1;
        mac1=[zeros(n+m) zeros(n+m) -Cfal'*Npom'*B1';...
           zeros(n+m) zeros(n+m) -Cfal'*Npom'*B2';...
           -B1*Npom*Cfal -B2*Npom*Cfal zeros(n+m)];
      wierzcholek1=wierzcholek1+NullSpace(j,i)*mac1;
   end

  for j=1:3*(n+m)
        wiersz=wierzcholek1(j,:);
       fprintf(fid,'%i ',wiersz(1,1:3*(n+m)));
       fprintf(fid,'\n');
  end
  for j=1:(n+m)
        wiersz=wierzcholek2(j,:);
        fprintf(fid,'%i ',wiersz(1,1:n+m));
       fprintf(fid,'\n');
  end
  for j=1:(n+m)
        wiersz=wierzcholek3(j,:);
       fprintf(fid,'%i ',wiersz(1,1:n+m));
       fprintf(fid,'\n');
  end
    fprintf(fid,'0 \n \n');
end

  mac=-eye(3*(n+m));
  for j=1:3*(n+m)
        wiersz=mac(j,:);
        fprintf(fid,'%i ',wiersz(1,1:3*(n+m)));
        fprintf(fid,'\n');
  end
  mac=-eye(n+m);
  for jjj=1:2
  for j=1:(n+m)
        wiersz=mac(j,:);
        fprintf(fid,'%i ',wiersz(1,1:(n+m)));
       fprintf(fid,'\n');
   end
    end
  fprintf(fid,'1 \n \n');
  fclose(fid);
  ex1=getlmis;
toc

nazwa_mat=['stabilizacja2D_out_bezX' num2str(n) 'X' num2str(r) 'X' num2str(m)]
eval(['save '  nazwa_mat])
```

Uniwersytet Zielonogórski
Wydział Elektrotechniki, Informatyki i Telekomunikacji

Aspekty obliczeniowe analizy i syntezy procesów powtarzalnych

Streszczenie

mgr inż. Bartłomiej Sulikowski
Promotor: prof. Krzysztof Gałkowski

Przez wiele ostatnich lat zostało zdefiniowanych wiele problemów, które, pomimo tego, że istniały warunki teoretyczne umożliwiające ich rozwiązanie, z praktycznej strony były nierozwiązywalne. Obecnie, pewna część tych problemów może być rozwiązywana dzięki możliwościom, jakie dało opracowanie efektywnych algorytmów numerycznych oraz dostarczenie ogromnej mocy obliczeniowej, jaką zapewniają nowoczesne komputery (wliczając do tej grupy klastry komputerowe). Problemy analizy i syntezy skomplikowanych wielowymiarowych układów dynamicznych zaliczają się do wspomnianej podklasy problemów praktycznie nierozwiązywalnych. Pomimo tego, że istnieją wyniki teoretyczne, ze względu na brak efektywnych metod, wyniki te pozostawały w sferze teorii. Problemy rozważane w niniejszej dysertacji odnoszą się do komputerowo wspomaganego rozwiązywania zadań analizy i syntezy specjalnej podklasy układów wielowymiarowych, tj. Liniowych Procesów Powtarzalnych. W rozprawie przedstawiono wyniki odnoszące się do tego w jaki sposób metody mające źródła w dziedzinie Informatyki można zastosować w celu rozwiązania postawionych zadań. Do takich metod zalicza się wyniki przedstawione w rozprawie, tj. opracowanie podejścia iteracyjnego przy rozwiązywaniu zadań syntezy oraz opracowanie metodologii umożliwiającej wykorzystanie bardzo efektywnych numerycznie algorytmów optymalizacji wypukłej do rozwiązywania postawionych problemów.

Układy wielowymiarowe (nD) stanowią uogólnioną formę klasycznych układów 1D. W ciągu ostatnich lat są obiektem zainteresowania badaczy zarówno z teoretycznego, jak i praktycznego punktu widzenia (np. [1, 2, 3, 4, 5, 6, 7]). Mogą one znajdować zastosowanie wszędzie tam, gdzie klasyczne podejście bazujące na modelu jednowymiarowym nie daje satysfakcjonujących rezultatów.

Ogólnie, układy nD w porównaniu z 1D charakteryzują się tym, że w opisie systemu występuje więcej niż jedna zmienna niezależna. W klasycznych układach 1D, zmienna niezależna na ogół ma charakter czasowy. W układach 2D (lub ogólnie nD) możemy mówić o czasie w postaci wektora 2 (n)-elementowego, bądź jedna ze zmiennych ma charakter czasowy, a pozostałe charakter przestrzenny. Najczęściej rozpatrywanymi układami 2D są systemy opisane dwuwymiarowym modelem Roessera (RM) [21] lub modelem Fornasiniego-Marchesiniego (FM) [22]; niemniej jednak zostały również zdefiniowane inne modele układów 2D (nD), pochodnych od dwóch wymienionych powyżej.

Układy wielowymiarowe nie są jedynie modelami teoretycznymi. Znajdują one zastosowanie przy modelowaniu procesów z dziedzin: Automatyki, Informatyki, Telekomunikacji, Akustyki, Elektrotechniki, Elektroniki i innych dziedzin techniki. Jako kilka szczegółowych aplikacji można przytoczyć: wielowymiarowe filtrowanie [8, 9], kodowanie/dekodowanie sygnałów z wykorzystaniem technik nD [10], przetwarzanie obrazów [11, 12], wielowymiarowe przetwarzanie sygnałów [13, 14, 15, 16] i wiele innych.

W niniejszej dysertacji rozpatrywana jest szczególna podklasa układów 2D. Układy te są nazywane Liniowymi Procesami Powtarzalnymi (*ang. Linear Repetitive Processes* - LRP) [17, 18, 19, 20] i różnią się one tym od podstawowych układów 2D (RM lub FM), że jedna ze zmiennych niezależnych odnosząca się do dynamiki jest ograniczona.

Procesy powtarzalne opisują ciąg wykonań pewnej czynności. Stąd, wspomniana zmienna ograniczona stanowi znacznik czasu bądź pozycji w trakcie bieżącego wykonania systemu. Zmienna ta może mieć charakter dyskretny (mówimy wtedy o dyskretnych LRP) lub ciągły (różniczkowe LRP). Druga ze zmiennych niezależnych pojawiających się w modelu stanowym LRP oznacza numer bieżącego wykonania „całej" czynności, iteracji, bądź, jak to jest określane w kontekście procesów powtarzalnych, pasa. Naturalne jest, że zmienna ta przyjmuje zawsze wartości dyskretne. Zatem, aby uściślić, kiedy rozważamy procesy powtarzalne, mówimy o dynamice wzdłuż pasa (zmienna ograniczona, ciągła lub dyskretna) i dynamice z pasa na pas (dyskretna zmienna nieograniczona).

Procesy fizyczne, które mogą być z powodzeniem modelowane jako LRP zawiera poniższa lista: walcowanie metalu [23, 17, 24, 25], wydobywanie węgla [17, 20], iteracyjne sterowanie z uczeniem (ILC) [26, 27, 28, 29], rozwiązywanie problemów występujących przy rozwiązywaniu zadania optymalnego sterowania obiektami nieliniowymi [30, 31], przestrzennie połączone podsystemy [32] i inne. Lista i opis wybranych aplikacji fizycznych znajduje się w Rozdziale 2 niniejszej rozprawy.

Należy tu zauważyć, że własności rozważanej klasy układów dynamicznych, tj. Liniowych Procesów Powtarzalnych mają ścisłe powiązania z własnościami algorytmów iteracyjnych. Jedną ze wspomnianych praktycznych aplikacji, które z powodzeniem mogą być modelowane jako LRP, są zadania dotyczące iteracyjnego sterowania z uczeniem. Zatem zapewnienie stabilności procesu powtarzalnego, odnosi się do zapewnienia zbieżności procesu iteracyjnego, który jest modelowany jako LRP. Co więcej, przedstawione w pracy schematy syntezy, zapewniające dodatkowe, oprócz stabilności, własności układu w zamkniętej pętli sprzężenia zwrotnego (takie jak marginesy stabilności lub osiągnięcie żądanej postaci modelu), w odniesieniu do zadań iteracyjnego sterowania z uczeniem (*ang. Iterative Learning Control - ILC*), odnoszą się do „polepszenia" zachowania procedury iteracyjnej (np. szybszej minimalizacji błędu śledzenia, braku oscylacji błędu itp.). Stąd można stwierdzić, że istnieją silne związki pomiędzy własnościami systemu dynamicznego (teoria systemów), a własnościami iteracyjnej procedury obliczeniowej (informatyka).

Modele stanowe Liniowych Procesów Powtarzalnych

W trakcie ostatnich lat, w zależności od potrzeb, zostało zdefiniowanych wiele modeli procesów powtarzalnych w literaturze. W niniejszej dysertacji rozpatrywane są następujące modele Liniowych Procesów Powtarzalnych

- dyskretny model "podstawowy"w postaci (2.9)-(2.10) z warunkami brzegowymi (początkowymi) w postaci (2.11)
- różniczkowy model "podstawowy"w postaci (2.14)-(2.15) z warunkami brzegowymi (początkowymi) w postaci (2.16)

- dyskretny model "uogólniony" w postaci (2.19)-(2.20) z warunkami brzegowymi (początkowymi) w postaci (2.11)

Do pozostałych modeli LRP zdefiniowanych w literaturze należą: modele osobliwych wzdłuż pasa procesów powtarzalnych (dyskretne [57] lub różniczkowe [58]), dyskretny model rozszerzony [55] oraz dyskretny model falowy [56].

Przy badaniu własności procesów powtarzalnych często korzysta się z tzw. równoważnego modelu 1D, opisanego równaniami (2.24)-(2.25), z macierzami określonymi przez (2.26) – dla modelu "podstawowego" (2.9)-(2.10) oraz przez (2.27) – dla modelu "uogólnionego" (2.19)-(2.20). Należy tu zauważyć, że postać ta jest możliwa do otrzymania tylko dla dyskretnych LRP. Przedstawienie LRP w postaci modelu 1D jest możliwe dzięki temu, że zmienna odnosząca się do dynamiki wzdłuż pasa jest ograniczona. Tworzy się go poprzez zastosowanie operacji przesunięcia zmiennych, pogrupowania wszystkich wektorów stanu, wyjścia i wejścia funkcjonujących na danym pasie w tzw. superwektory, a następnie określeniu równań 2D rozważanego LRP dla wszystkich możliwych pkt. na pasie i zebraniu ich do razem tak, aby możliwe było zapisanie ich wszystkich z wykorzystaniem otrzymanych wcześniej superwektorów. W związku z tym, równoważny model 1D opisuje tylko dynamikę w kierunku z pasa na pas (po k), a dynamika wzdłuż pasa (po p) jest niejako „ukryta" „wewnątrz" macierzy modelu. Ze względu na to, że równoważny model 1D traktuje całe pasy wykonania danej czynności jako pojedyncze wektory, jego rozmiary (rozmiary macierzy) są ściśle związane z długością pasa i zazwyczaj przyjmują bardzo duże wartości. Operacje przekształcania macierzy o potencjalnie ogromnych rozmiarach mogą powodować trudności z numerycznego pkt. widzenia. Ze względu na to, że równoważny model 1D jest modelem jednowymiarowym, większość metod analizy/syntezy opracowanych dla tej klasy systemów dynamicznych, może być stosowana dla równoważnego modelu 1D. Niemniej jednak, należy pamiętać, że ze względu na uproszczenie jakie wprowadza zastosowanie modelu 1D (brak uwzględnienia dynamiki wzdłuż pasa), model ten nadaje się do badania własności procesów powtarzalnych jedynie w ograniczonym zakresie.

Jak przedstawiono, ze względu na ułatwienia jakie można uzyskać przy rozwiązywaniu zadań analizy i syntezy dyskretnych (przy rozważanej własności stabilności asymptotycznej) procesów powtarzalnych, stosowany jest równoważny model 1D procesu powtarzalnego w postaci (2.24)-(2.25) z odpowiednio zdefiniowanymi macierzami modelu.

Istniejące warunki analizy i syntezy LRP i ograniczenia przy ich stosowaniu

Podstawową własnością systemów dynamicznych jest stabilność. Dla całej klasy procesów powtarzalnych można zdefiniować kilka jej rodzajów, z których najczęściej rozpatrywanymi są: stabilność asymptotyczna (Definicja 2.1) oraz stabilność wzdłuż pasa (Definicja 2.2) przedstawione w Podrozdziale 2.4 [17, 18].

Zapewnienie stabilności asymptotycznej dla procesu powtarzalnego gwarantuje, że po przejściu odpowiednio dużej liczy pasów, profil pasa będzie zbliżony do pewnego profilu granicznego, czyli proces nie będzie rozbieżny w kierunku z pasa na pas. Własność ta jednak nie gwarantuje, że uzyskany profil graniczny będzie stabilny w sensie klasycznego układu 1D. Stabilność profilu granicznego jest zagwarantowana przez „mocniejszą" własność, tj. stabilność wzdłuż pasa.

Warunek służący do sprawdzania stabilności asymptotycznej dla LRP w postaci (2.9)-(2.10) i (2.14)-(2.15) przedstawiono w Twierdzeniu 2.1. Niestety tego warunku nie da się zastosować dla uogólnionych procesów powtarzalnych (2.19)-(2.20). W celu sprawdzenia stabilności dla tej podklasy procesów powtarzalnych wykorzystuje się równoważny model 1D i warunek określający stabilność asymptotyczną przedstawia Twierdzenie 2.2. Warunki te bazują na sprawdzeniu czy wartość promienia spektralnego macierzy systemowej rozpatrywanego modelu jest mniejsza od 1. W tym miejscu pojawia się potencjalna trudność, tj. kiedy rozpatrywany jest równoważny model 1D, to rozwiązanie problemu wymaga dokonywania obliczeń na macierzach o ogromnych rozmiarach, co może być powodem błędów numerycznych, a nawet uniemożliwić wykonanie zadania analizy/syntezy. Należy tu nadmienić, że wymienione warunki są warunkami koniecznymi i wystarczającymi.

Jak wspomniano, stabilność wzdłuż pasa jest własnością „mocniejszą" od stabilności asymptotycznej. Zapewnienie jej gwarantuje, że profil pasa granicznego traktowanego jako układ 1D, jest stabilny. Można również zauważyć, że stabilność asymptotyczna jest jedną z własności koniecznych, wymaganych do zapewnienia stabilności wzdłuż pasa. Warunki określające stabilność wzdłuż pasa dla procesów (2.9)-(2.10) i (2.14)-(2.15) przedstawiono odpowiednio w Twierdzeniach 2.4 i 2.5. Dla procesów uogólnionych ta własność nie była rozważana.

Należy tu zaznaczyć, że stabilność wzdłuż pasa jest odpowiednikiem stabilności asymptotycznej ogólnego układu 2D, zapisanego w postaci modelu Roessera lub Fornasiniego-Marchesiniego. Istnieje kilka warunków służących do sprawdzania stabilności wzdłuż pasa procesów powtarzalnych. Można tego dokonać korzystając z warunków bazujących na badaniu wielomianu charakterystycznego rozważanego systemu (Twierdzenia 2.3, 2.4 i 2.5). Wspomniane warunki analizy są warunkami koniecznymi i wystarczającymi. Jednak w każdym z przypadków, praktyczne zastosowanie tych testów jest bardzo utrudnione, nawet dla problemów o niezbyt dużych wymiarach, a w przypadkach „nieakademickich", w większości przypadków niemożliwe. W pracy [41] pokazano, że analiza i synteza układów 2D z wykorzystaniem warunków bazujących na funkcji transmitancji lub wielomianie charakterystycznym z algorytmicznego punktu widzenia mogą być traktowane jako zadania należące do klasy problemów $\mathcal{NP}$-trudnych. Wynika to z faktu, że wymienione warunki wymagają sprawdzenia wartości numerycznej, jaką przyjmuje stosowany warunek dla nieskończenie dużego zbioru możliwości. Uściślając, zgodnie z najprostszym numerycznie istniejącym testem tego typu, opartym na Twierdzeniu 2.3, dla zbadania stabilności wzdłuż pasa, niezbędne jest zbadanie stabilności (w sensie Schura) pewnej macierzy zespolonej $G(z)$, dla każdej zespolonej liczby $z \in C$, leżącej na okręgu jednostkowym. W związku z tym, konieczne jest wykonanie nieskończenie wielu testów stabilności 1D, co jest zadaniem niewykonalnym, niezależnie od wymiaru problemu. Oczywiście, możliwe jest „zdyskretyzowanie" problemu, próbkując koło jednostkowe z pewną rozdzielczością. Podejście takie daje w efekcie skończoną liczbę testów, ale „cenę", jaką trzeba „zapłacić" jest to, że uzyskany test jest tylko konieczny, a więc w zasadzie nie ma pewności czy pomiędzy rozważanymi punktami na okręgu jednostkowym nie ma punktu, dla którego test nie jest spełniony. Rzecz jasna, że wykonanie większej liczby testów („gęstsze" próbkowanie okręgu), daje większą pewność uzyskania prawidłowego wyniku. Syntuacja jeszcze bardziej się komplikuje przy zagadnieniach syntezy.

Podsumowując, zarówno przy rozważaniu stabilności asymptotycznej, jak i stabilności wzdłuż pasa, są dostępne warunki konieczne i wystarczające, mogące służyć do sprawdzania tych własności jednak, ich stosowalność jest ograniczona, z powodu:

- dla stabilności asymptotycznej - ogromnych rozmiarów rozwiązywanych zadań,
- dla stabilności wzdłuż pasa - faktu, że w taki sposób sformułowane zadanie należy do klasy problemów $\mathcal{NP}$-trudnych.

Teza pracy

W związku z opisanymi problemami, występującymi przy rozwiązywaniu zadań analizy/syntezy Liniowych Procesów Powtarzalnych, postawiono następującą tezę:

> ***Możliwe jest opracowanie efektywnych numerycznie metod analizy i syntezy złożonych procesów powtarzalnych (o wielkich wymiarach) przy zastosowaniu nowoczesnych pakietów numerycznych LMI, wzmocnionych użyciem technik obliczeń równoległych.***

Celem niniejszej rozprawy jest udowodnienie powyższej tezy.

Zastosowanie Liniowych Nierówności Macierzowych przy zadaniach analizy i syntezy układów dynamicznych

Jako alternatywa dla istniejących warunków przedstawionych w poprzednim podrozdziale, pojawiła się możliwość zastosowania Liniowych Nierówności Macierzowych (zobacz np. [42, 44]) do rozwiązywania problemów analizy i syntezy procesów powtarzalnych. W Rozdziale 3 zawarto opis wszystkich aspektów związanych z LMI i zastosowaniem ich rozwiązywaniu zadań analizy/syntezy procesów dynamicznych. Wiadomości podstawowe jak stosować LMI przy badaniu klasycznych układów 1D, zostały zawarte w następujących publikacjach [42, 43, 44, 45, 80, 81].

Większość rozważanych zadań analizy i syntezy układów dynamicznych nie jest zdefiniowane w postaci Liniowych Nierówności Macierzowych. Niemniej jednak korzystając z przekształceń pochodzących z dziedziny algebry liniowej i teorii nierówności, zadania te można przekształcić do postaci LMI. Kluczowym przekształceniem stosowanym przy przeformułowaniu zadań do poprawnej postaci LMI, przedstawionym w niniejszej rozprawie (Podrozdział 3.3) jest tzw. uzupełnienie Schura [42, 87, 88]. Innymi wykorzystywanymi przekształceniami nierówności są m. in. zamiana zmiennych lub operacja kongruencji.

Dodatkowo, w rozdziale 3 przedstawiono możliwość wykorzystania pakietów programowych (*ang. solver*) do rozwiązywania zadań programowania Pół-Określonego (*ang. SemiDefinite Programming* - SDP) [101]. Przedstawienie warunków LMI w postaci SDP umożliwia wykorzystanie dowolnego z wielu dostępnych solverów SDP, wliczając w to pakiet obliczeń równoległych SDPARA [98, 97], co znacząco rozszerza możliwości obliczeniowe. Należy zauważyć, że w przypadku, kiedy rozwiązanie postawionego zadania analizy/syntezy w postaci LMI wymaga zastosowania pakietu numerycznego, służącego pierwotnie do rozwiązywania zadań SDP, konieczne jest przeformułowanie postawionego zadania LMI do rozwiązywalnej formy SDP. A zatem pierwotny warunek w postaci LMI, polegający na sprawdzeniu istnienia rozwiązania (*ang. feasibility*), jest formułowany jako zadanie optymalizacji wypukłej i tak rozwiązywany. W Podrozdziale 3.5 przedstawiono procedurę, jaką należy przeprowadzić w celu przedstawienia warunku LMI w postaci ogólnej SDP.

Podstawową zaletą zastosowania metod LMI przy komputerowo wspomaganym rozwiązywaniu zadań analizy i syntezy jest wysoka efektywność numeryczna tych algorytmów. Warunki w postaci LMI charakteryzują się tym, że rozwiązanie zadanego problemu jest otrzymywane w czasie wielomianowym w stosunku do rozmiaru rozważanego problemu. Tak wysoka efektywność jest uzyskiwana dzięki zastosowaniu algorytmów punktu wewnętrznego (*ang. Interior Point Method* - IPM) [100]. O ile idea algorytmu IPM bazuje na algorytmie nieliniowej optymalizacji Newtona, to w literaturze (zob. np. [99, 100, 115, 98, 97, 85]) można znaleźć wiele wersji tej metody. Algorytmy IPM charakteryzują się tym, że podczas poszukiwania rozwiązania możliwe wykonanie części operacji równolegle, dzięki czemu znacząco wzrasta prędkość obliczeń. Ma to szczególne znaczenie w sytuacjach kiedy rozwiązywane zadania są wielkich rozmiarów. W pracy (Podrozdział 3.7) szczegółowo opisano algorytm zaimplementowany w pakiecie SDPA [98, 97]. Jak wspomniano, możliwe jest źrównoleglenie"tego algorytmu. Obliczenie przybliżenia odwrotności macierzy Hesjanu, która jest wykorzystywana przy określaniu kolejnego przybliżenia poszukiwanego rozwiązania jest operacją, która może być z powodzeniem wykonywana równolegle. Uściślając, kolejne wiersze tej macierzy mogą być obliczane na różnych procesorach. Szczegóły tego procesu można znaleźć w Podrozdziale 3.8.

Rozdział 3 zawiera listę dostępnych pakietów służących do rozwiązywania LMI/SDP oraz opisy wybranych pakietów tj. MATLAB LMI CONTROL TOOLBOX [44], SCILAB LMI OPTIMIZATION PACKAGE [121], SDPARA.

Analiza i synteza LRP z zastosowaniem LMI

Rozdział 4 zawiera warunki w postaci LMI, umożliwiające badanie (analiza) obu rodzajów stabilności i stabilizację (synteza) rozważanych modeli LRP. Oprócz podstawowych warunków odnoszących się do analizy/syntezy, rozważany był szereg dodatkowych aspektów. Dotyczą one: obliczania marginesów stabilności - Podrozdział 4.5, „dopasowania modelu” (zarówno w wersji dla równoważnego modelu 1D - Podrozdział 4.10.1, jak i 2D - Podrozdział 4.10.2), „odseparowanie” dynamiki - Podrozdział 4.7, zaproponowania algorytmu sukcesywnej stabilizacji - Podrozdział 4.8 oraz zaproponowania warunków stabilizacji rozważanego systemu wykorzystując jedynie informacje dostępne w wyjściu systemu (bez informacji z wektorów stanu) - Podrozdział 4.9.

Podstawowym powodem motywującym zastosowanie metod LMI przy rozwiązywaniu zadań analizy i syntezy jest to, że wykorzystują one bardzo efektywny numerycznie algorytm, umożliwiający znajdowanie rozwiązania. A zatem, dzięki LMI możliwe jest określenie warunków, które dają się efektywnie rozwiązywać. Jak wiadomo, algorytm IPM zaimplementowany w solverach LMI, należy do klasy $\mathcal{P}$. Przy badaniu stabilności asymptotycznej umożliwia to relatywnie szybko otrzymać wynik. W przypadku stabilności wzdłuż pasa, dzięki zastosowaniu LMI, możliwe jest przekształcenie zadań z klasy $\mathcal{NP}$-trudnych, do rozwiązywalnych zadań z klasy $\mathcal{P}$ (przy jednoczesnej redukcji z warunków koniecznych i wystarczających, na warunki wystarczające).

Wyniki zaprezentowane w Rozdziale 4 są w większości oryginalnymi osiągnięciami autora i zostały (bądź wkrótce zostaną) opublikowane na forum krajowym i międzynarodowym.

Stabilność asymptotyczna

Jak wspomniano, zapewnienie stabilności asymptotycznej gwarantuje istnienie tzw. pasa granicznego, tj. przebiegu wyjściowego, który przy kolejnych pasach ma taki sam kształt. W przypadku modeli „podstawowych" LRP (dyskretnego (2.9)-(2.10) i różniczkowego (2.14)-(2.15)) z najprostszymi warunkami brzegowymi, analiza jest stosunkowo prosta i nie generalnie nie przysparza problemów numerycznych. Bazuje ona na badaniu spektrum macierzy D_0. Zatem, stosowanie metod LMI nie jest w tym przypadku analizy uzasadnione. Niemniej jednak, kiedy rozważana jest synteza tych procesów, okazuje się, że metody LMI dostarczają naturalnego sposobu w jaki, od analizy (badanie stabilności) przejść do zadania syntezy (stabilizacji).

Sytuacja diametralnie zmienia się kiedy pod badane są uogólnione dyskretne modele LRP w postaci (2.19)-(2.20). Dla tej klasy układów własność stabilności asymptotycznej określono wykorzystując postać równoważnego modelu 1D w postaci (2.24)-(2.25). Dla tej podklasy układów istnieje również test spektralny, ale polega on na obliczeniu wartości promienia spektralnego macierzy systemowej, występującej w równoważnym modelu 1D (Twierdzenie 2.2). Zauważyć tu należy, że tak zdefiniowane zadanie analizy, ponownie można traktować jako trudne do przekształcenia do zadania syntezy. Ponadto, zarówno w przypadku analizy, jak i syntezy, pojawiają się problemy związane z dużymi rozmiarami macierzy, które są przetwarzane. Dlatego też, w tym przypadku metody LMI umożliwiają uzyskanie rozwiązania w czasie wielomianowym do rozmiaru zadania, a dodatkowo dostarczają prostego sposobu przejścia od analizy do syntezy (wyznaczenia parametrów pętli sprzężenia zwrotnego – stabilizacji układu niestabilnego).

W Podrozdziale 4.1 przedstawiono warunki w postaci LMI służące do badania stabilności asymptotycznej. Twierdzenie 4.1 odnosi się do analizy „podstawowych" LRP, a Twierdzenie 4.2 – do analizy uogólnionych dyskretnych LRP (z wykorzystaniem równoważnego modelu 1D).

Podrozdział 4.2 zawiera warunki mogące służyć do rozwiązywania zadań syntezy (Twierdzenia 4.3 i 4.4).

Należy tutaj podkreślić, że zadania analizy/syntezy LRP z wykorzystaniem równoważnego modelu 1D, wymagają przetwarzania macierzy o potencjalnie ogromnych rozmiarach. Tak duże rozmiary wynikają z faktu, iż stosowany równoważny model 1D jest modelem dużego (ogromnego) rzędu, a operacje z wykorzystaniem jego macierzy mogą się wiązać z ryzykiem powstawania błędów numerycznych. Dodatkową przeszkodą przy stosowania warunków wykorzystujących równoważny model 1D jest to, że przechowywanie i obliczenia przeprowadzane na tak ogromnych macierzach mogą być wykonywane bardzo powoli, bądź pojedynczy komputer może nie posiadać wystarczająco dużo zasobów (pamięci operacyjnej, mocy obliczeniowej) do wykonania obliczeń. Przeszkody te mogą być ominięte na kilka sposobów. Jednym z nich jest wyeksploatowanie mocy obliczeniowej jaką dostarczają klastry. Innymi są: wykorzystanie możliwości uproszczenia zadania syntezy poprzez uproszczenie spektrum macierzy systemowej równoważnego modelu 1D lub zastosowanie procedury iteracyjnej do rozwiązania zadania syntezy.

Zastosowanie klastrów przy analizie i syntezie LRP

Dzięki zastosowaniu klastrów możliwe jest podjęcie prób rozwiązania zadań, które charakteryzują się wielkimi wymiarami. Wykorzystanie dużej mocy obliczeniowej dostarczonej przez klastry, wydaje się więc naturalnym sposobem na usprawnienie procesu obliczeniowego.

W celu rozwiązania zadania syntezy LRP przy wykorzystaniu klastrów obliczeniowych, konieczne jest przekształcenie zadania w postaci LMI do odpowiedniego zadania w postaci SDP. Wynika to z faktu, że nie ma dostępnych pakietów oprogramowania służących do rozwiązywania zadań w postaci LMI. Dopiero takie zadanie może być rozwiązywane z wykorzystaniem pakietów oprogramowania dostępnych dla klastrów. W celu przeformułowania zadania z LMI do SDP zaimplementowano odpowiednie funkcje pracujące w środowisku MATLAB (wybrane z nich zostały przedstawione w Załączniku B.3). Wyniki numeryczne odnoszące się do wykorzystania klastrów, przedstawione w poniższej pracy, uzyskane zostały dzięki zastosowaniu pakietu SDPARA [98, 97].

W przykładzie 4.2 rozważano rozwiązywanie zadań syntezy wielkowymiarowych procesów powtarzalnych z wykorzystaniem klastrów. Pokazano, że wykorzystanie technik obliczeń równoległych znacząco przyspiesza uzyskanie rozwiązania, a także umożliwia syntezę procesów o dużych rozmiarach. W tabeli 4.1 przedstawiono czasy potrzebne do rozwiązania procesu syntezy przy wykorzystaniu różnych liczb węzłów w klastrze. Na komentarz zasługuje wynik, jaki został otrzymany dla największego z rozważanych procesów (liczba zmiennych stanu $n = 5$, liczba wyjść $m = 3$, liczba wejść $r = 1$, liczba pkt. na pasie $\alpha = 40$), którego macierze równoważnego modelu 1D: systemowa Φ i wejścia Δ miały rozmiary, odpowiednio 120×120 i 120×40. Poprzez tak określone zadanie, rozwiązanie było poszukiwane w przestrzeni R^{12060} i zostało wyznaczone w czasie ok. 25 minut.

Dodatkowymi aspektem badanym przy tej okazji, było oszacowanie jaką złożonością obliczeniową charakteryzuje się stosowany algorytm IPM. Wiadomo, że zastosowanie metod LMI/SDP dostarcza rozwiązania w czasie wielomianowym w stosunku do rozmiaru rozpatrywanego problemu. Dla wykonanych przykładów (rozmiary zadań od 126 do 12060 zmiennych) okazało się, że złożoność obliczeniową najlepiej przybliża funkcja kwadratowa (Rysunek 4.1).

Ciekawym aspektem związanym z zastosowaniem klastrów jest przyspieszenie obliczeń w stosunku do liczby węzłów używanych w klastrze. Uzyskane wyniki pokazują, że dla największego z rozważanych zadań tj. dla 12060 zmiennych, zastosowanie 16 węzłów w klastrze spowodowało przyspieszenie rzędu 3.5 raza w stosunku do rozwiązania uzyskanego z wykorzystaniem 1 węzła (Rysunek 4.3). Co ciekawe, w przypadku małych problemów (500 i mniej zmiennych decyzyjnych) stosowanie klastrów nie jest uzasadnione - w takich sytuacjach szybciej działają pojedyncze komputery (Rysunek 4.4 a)). Sytuacja ta wynika z faktu, że dla tak małych zadań więcej czasu jest „tracone” na rzecz synchronizacji (rozdzielenie obliczeń, zebranie wyników) obliczeń równoległych, niż „zyskiwane” dzięki zrównolegleniu obliczeń.

Podsumowując, przeprowadzone badania pokazują, że klastry obliczeniowe usuwają ograniczenia jakim poddane są pojedyncze komputery i doskonale mogą być stosowane przy analizie i syntezie procesów powtarzalnych. Osiągnięte wyniki świadczą, że wybrane oprogramowanie może być z sukcesem stosowane, a liczba zmiennych optymalizacyjnych/decyzyjnych, określająca rozmiar przestrzeni poszukiwań rozwiązania, może swobodnie przekraczać 10000. Co więcej, jest ona ograniczona głównie przez rozmiar pamięci RAM dostępnej na klastrze.

Odseparowanie dynamik jako analityczna metoda uproszczenia syntezy procesów powtarzalnych

Innym sposobem „poradzenia sobie” z wielką wymiarowością zadań syntezy równoważnego modelu 1D jest zastosowanie podejścia analitycznego. Podejście zostało przedstawione w Podrozdziale 4.7 i polega ono na wprowadzeniu etapu preprocessingu do zadania syntezy. Ten wstępny etap ma za zadanie doprowadzić do znaczącego uproszczenia spektrum macierzy systemowej Φ

równoważnego modelu 1D. Jego rezultatem jest otrzymanie Φ, której wartości własne wynoszą $m(\alpha - 1)$ zer i dodatkowe wartości własne są równe spektrum macierzy wyrażonej jako $\sum_{j=1}^{\alpha-1} D_j$. Dzięki takiemu przedstawieniu zadania możliwe jest przeprowadzenie syntezy dla znacznie zredukowanego Φ, tj. zamiast obliczać parametry sprzężenia zwrotnego dla $\Phi \in \mathcal{R}^{m\alpha \times m\alpha}$, oblicza się je dla macierzy z przestrzeni $\mathcal{R}^{m \times m}$. Zamieszczone przykłady ilustrują działanie prezentowanego podejścia.

Wykorzystanie zalet podejścia iteracyjnego w rozwiązywaniu zadania syntezy LRP. Algorytm sukcesywnej stabilizacji

Algorytm sukcesywnej stabilizacji jest kolejnym podejściem mającym na celu ominięcie przeszkód związanych z wielką wymiarowością zadań syntezy przy korzystaniu z opisu LRP w postaci równoważnego modelu 1D. Opracowany algorytm został zaprezentowany w Podrozdziale 4.8. Jego idea polega na wykorzystaniu zalet, jakimi charakteryzują się procedury iteracyjne, stosowane przy rozwiązywaniu skomplikowanych numerycznie zadań.

W skrócie, zasadę działania algorytmu sukcesywnej stabilizacji można przedstawić w następujących krokach:

Krok 1 Wyznaczenie parametrów sprzężenia zwrotnego dla procesu o mniejszej, niż oryginalna, długości pasa. Sprawdzenie, czy taka postać sprzężenia zwrotnego zapewnia stabilność dla oryginalnej długości pasa.

Krok 2 Jeśli nie - wydłużenie pasa o założoną wartość interwału i wykonanie operacji z *Kroku 1.*

Co więcej, przy wykorzystywaniu zaproponowanego schematu syntezy, możliwe jest spełnienie dodatkowego założenia dotyczącego zachowania struktury macierzy procesu powtarzalnego w zamkniętej pętli sprzężenia zwrotnego.

Zamieszczone przykłady ilustrują działanie zaproponowanego algorytmu.

Stabilność wzdłuż pasa

Drugim rodzajem badanej w rozprawie stabilności, była stabilność wzdłuż pasa. Jak wspomniano, własność ta jest „mocniejsza" od stabilności asymptotycznej (w rzeczy samej, zapewnienie stabilności asymptotycznej stanowi jeden z warunków koniecznych wymaganych do zapewnienia stabilności wzdłuż pasa - Twierdzenia 2.4 i 2.5). Własność ta charakteryzuje się tym, że jej zapewnienie powoduje, że istniejący profil pasa granicznego, traktowany jako klasyczny układ jednowymiarowy (z dynamiką funkcjonującą w kierunku wzdłuż pasa), jest stabilny.

Jak to zostało przedstawione, istniejące wyniki teoretyczne odnoszące się do analizy przy stabilności wzdłuż pasa, mogą być traktowane jako zadania $\mathcal{NP}$-trudne. Dlatego też w praktyce, ich stosowanie jest w zasadzie niemożliwe. Dodatkowym problemem, związanym z istniejącymi warunkami jest fakt, że nie dostarczają one praktycznego podejścia, w jaki sposób, od analizy (badanie stabilności) przejść do rozwiązywania zadania syntezy (stabilizacji).

W Rozdziale 3 pokazano, że dzięki zastosowaniu metod badania stabilności opartych na podejściu Lyapunova, które mogą prowadzić do równoważnych warunków w postaci LMI, możliwe jest zdefiniowanie warunków, mogących służyć do analizy i syntezy, procesów powtarzalnych.

„Ceną” jaką trzeba „zapłacić” za stosowanie Liniowych Nierówności Macierzowych, w odniesieniu do stabilności wzdłuż pasa, jest to, że uzyskane warunki są tylko wystarczające, co jest charakterystyczne dla układów 2D.

Należy tu nadmienić, że w pracy [52], zaprezentowano sposób, w jaki można zmniejszyć konserwatywność stosowanego warunku LMI odnoszącego się do analizy układów 2D i docelowo „zbliżyć go” do warunku koniecznego i wystarczającego. Podejście to powoduje jednak znaczące zwiększenie poziomu skomplikowania rozważanych nierówności macierzowych i ich rozmiarów, co znacząco ogranicza sensowość ich użycia. W związku z tym, warunki LMI oparte o to podejście nie są rozważane w niniejszej rozprawie.

Porównanie istniejących warunków LMI

Istnienie dwóch postaci warunków LMI służących do badania stabilności wzdłuż pasa (analizy) i stabilizacji (syntezy) dyskretnych LRP, jest związane z faktem, że rozważany proces powtarzalny można przedstawić w postaci znanych modeli stanowych Roessera (RM) [21] lub Fornasiniego-Marchesiniego (FM)[22]. Dlatego też, możliwe jest „przeniesienie” warunków LMI określonych dla RM i FM, dla badanej klasy LRP. Twierdzenia 4.6 i 4.7 (Podrozdział 4.3) przedstawiają warunki LMI stosowane przy rozwiązywaniu zadania analizy, Twierdzenia 4.9 i 4.10 (Podrozdział 4.4) stanowią ich wersję „zamkniętą” i odnoszą się do rozwiązywania zadań syntezy.

Ze względu na istnienie dwóch podejść służących do analizy/syntezy, można pokusić się o porównanie ich. Wyniki takiego porównania, wykonanego z wykorzystaniem klastra, zostały przedstawione w pracy. Z numerycznego pkt. widzenia pierwsza różnica pomiędzy tymi warunkami, ujawnia się, kiedy sprawdzone jest jak wielu zmiennych decyzyjnych (rozmiar przestrzeni rozwiązań) poszukuje się przy ich rozwiązywaniu. Tabela 4.2 pokazuje, że zastosowanie warunku LMI z Twierdzenia 4.6 wymaga zaangażowanie prawie dwakroć więcej zmiennych niż warunek LMI z Twierdzenia 4.7. Naturalnym wydaje się zatem założenie, że rozwiązanie zadania przy stosowaniu Twierdzenia 4.6 będzie otrzymywane znacznie później niż przy zastosowaniu Twierdzenia 4.7. Przypuszczenia te potwierdza przykład 4.3.

Z drugiej strony, ze względu na to, że sprawdzane warunki są tylko wystarczające, porównywano ich konserwatywność, czyli sprawdzano, jak często warunki LMI pozwalały na rozwiązanie tego samego zadania. Okazało się, że przy zastosowaniu warunku z Twierdzenia 4.6, rozwiązanie postawionego zadania syntezy znajdowano znacznie częściej niż przy Twierdzeniu 4.7. W przykładzie 4.4 zaprezentowano model LRP, dla którego przy użyciu warunki z Twierdzenia 4.6 możliwe było dokonanie procesu syntezy. W tym samym przypadku zastosowanie Twierdzenia 4.7 zawodziło. Co ciekawe, przykład przeciwny, tj. kiedy Twierdzenie 4.7 działa, a Twierdzenie 4.6 nie działa, nie został znaleziony. Istnienie tak znacznej różnicy w funkcjonowaniu tych warunków może być wytłumaczone tym, że warunek z Twierdzenia 4.6 poszukuje rozwiązania w przestrzeni o rozmiarze prawie dwa razy większym niż Twierdzenie 4.7. A zatem to, co było atutem (mniejsza liczba zmiennych decyzyjnych) i prowadziło do szybszego uzyskania rozwiązania, tutaj jest przeszkodą.

W dalszej części prezentowane schematy syntezy, odnoszące się do stabilności wzdłuż pasa, bazują na warunku LMI przedstawionym w Twierdzeniu 4.6.

Badanie marginesów stabilności

Często rozwiązując zadania analizy/syntezy, oprócz stabilności wzdłuż pasa jako takiej, rozważa się dodatkowe aspekty. Do takiej grupy należy wyznaczanie (jako zadanie dodatkowe rozpatrywane przy analizie) i zapewnienie (jako zadanie dodatkowe rozpatrywane przy syntezie) tzw. marginesów stabilności. Dla rozważanej klasy procesów dynamicznych, marginesy stabilności zdefiniowano w Podrozdziale 4.5. Tam również zaprezentowano metody LMI umożliwiające obliczanie marginesów stabilności.

W Podrozdziale 4.6 przedstawiono schemat syntezy LRP z wykorzystaniem LMI zapewniającej fakt, że system w zamkniętej pętli sprzężenia charakteryzuje się założonymi marginesami stabilności.

Synteza z wykorzystaniem informacji zawartych w wyjściu LRP

Osobną klasą problemów jest synteza LRP z zastosowaniem tylko informacji zawartych w wektorze wyjściowym układu. Takie postawienie zadania jest umotywowane tym, że w praktyce często jedynymi informacjami dostępnymi pomiarowo są pochodzące z wyjścia układu. Z drugiej strony, prezentowane podstawowe warunki syntezy bazują na wykorzystaniu informacji zarówno z wyjścia jak i wektora stanu układu. Dlatego w sytuacji, kiedy wektor stanu układu nie jest dostępny (sytuacja taka może mieć miejsce stosunkowo często), możliwe jest zastosowanie dwóch podejść, tj. podjęcia próby estymacji wektora stanu (opracowanie obserwatora stanu) i wykorzystania uzyskanych informacji w znanych schematach syntezy, bądź opracowanie metodologii umożliwiającej zastosowanie metod LMI bezpośrednio (tzw. synteza „od wyjścia" układu).

W Podrozdziale 4.9 przedstawiono to ostatnie podejście zarówno dla procesów dyskretnych, jak i różniczkowych. Należy tu zauważyć, że rozważane warunki LMI umożliwiające przeprowadzenie syntezy „od wyjścia" procesu powtarzalnego charakteryzują się tym, że wprowadzają dodatkowe ograniczenia do warunku LMI, co powoduje zwiększenie jego konserwatywności. Jako remedium zaprezentowano możliwość rozszerzenia rozważanych schematów syntezy w celu zmniejszenia konserwatywności - Podrozdziały 4.9.1 i 4.9.2.

Ze względu na to, że przy rozwiązywaniu zadania syntezy „od wyjścia" następuje zwiększenie liczby zmiennych decyzyjnych, oraz, przy zastosowaniu jednego z podejść rozszerzonych, zaobserwować można drastyczne zwiększenie rozmiaru LMI, uzasadnione wydaje się użycie klastrów do rozwiązywania zadań tej klasy. Niestety, poważną przeszkodą jest tutaj fakt, że dla potrzeb założonego schematu syntezy, wymagane było wprowadzenie dodatkowego ograniczenia równościowego występującego przy warunku LMI. Konieczne jest zatem przeprowadzenie procedury przekształcenia tego ograniczenia do postaci ograniczenia nierównościowego (metodę takiego przeformułowania pokazano w Podrozdziale 3.6), a następnie rozwiązanie otrzymanego zadania. Po otrzymaniu rozwiązania, konieczny jest powrót do oryginalnego zadania i ostatecznie wyznaczenie parametrów pętli sprzężenia.

W załączniku B.3 dołączono funkcję Matlaba umożliwiającą przeformułowanie zadań syntezy „od wyjścia" i zapisywanie ich do pliku, w postaci rozwiązywalnej przez pakiet Sdpara.

Dopasowanie modelu

Kolejnymi rozważanymi schematami syntezy, są schematy mające oprócz zapewnienia stabilności (asymptotycznej bądź wzdłuż pasa), dodatkowo zapewnienie żądanej postaci układu w zamkniętej pętli sprzężenia zwrotnego. Techniki te nazwane zostały metodami dopasowania modelu (*ang. model matching*).

W kontekście wykorzystania równoważnego modelu 1D (Podrozdział 4.10.1), tak określone schematy syntezy, przedstawiają wyniki i działają podobnie jak znane z teorii klasycznych systemów 1D, techniki lokowania biegunów układu zamkniętego. Przy czym należy tutaj zauważyć, że techniki lokowania biegunów nie odnosiły się do zapewnienia dokładnej struktury układu zamkniętego, tylko do umieszczenia spektrum macierzy systemowej wewnątrz założonego obszaru płaszczyzny zespolonej. Zaproponowana metoda syntezy umożliwia „sprowadzenie" macierzy systemowej Φ do z góry założonej struktury. Przykład 4.14 prezentuje przykład działania zaproponowanej metody.

W przypadku systemów 2D, jak wiadomo nie ma opracowanych wyników odnoszących się do lokowania biegunów. Jest to związane z faktem, że bieguny układu 2D nie są pkt. na płaszczyźnie zespolonej, ale są prezentowane jako krzywe (zob. np. [50]). Stąd brak odpowiednich metod, odnoszących się do rozwiązania zadania lokowania biegunów zamkniętego układu 2D. W Podrozdziale 4.10.2 zaprezentowano metodę syntezy, która umożliwia uzyskanie zadanego modelu 2D, który może być uzyskany np. drogą symulacji, w zamkniętej pętli sprzężenia zwrotnego.

Sterowanie LRP przy zadanym profilu granicznym i odsprzężeniu zakłóceń

Ze względu na to, że procesy powtarzalne służą do modelowania procesów fizycznych, zapewnienie stabilności często oznacza jedynie pierwszy krok. Naturalnym następstwem jest wymaganie, aby rozważany system po odpowiedniej liczbie pasów osiągnął zadaną wartość wyjściową. Zatem w Rozdziale 5 założono opracowanie i przetestowanie metod sterowania umożliwiających osiągnięcie następujących celów

- stabilność (asymptotyczna lub wzdłuż pasa, w zależności od rozważanego problemu),
- osiągniecie zadanej wartości na wyjściu,
- usunięcie wpływu zakłóceń zewnętrznych oddziaływujących na proces powtarzalny.

Metodologię umożliwiającą osiągnięcie pierwszego z powyższych celów zaproponowano w Rozdziale 4. Zależnie od rozważanego problemu, zastosowano podstawowe warunki LMI służące do syntezy, dopasowania modelu i syntezy „od wyjścia" dla rozważanych procesów powtarzalnych. Dwa pozostałe cele zostały osiągnięte przez zaadaptowanie znanych dla układów 1D technik do wymagań układów 2D. Zostały tutaj przedstawione cztery podejścia służące osiągnięcia postawionych celów, tj. sterowanie bezpośrednie - Podrozdział 5.1, pośrednie - Podrozdział 5.2, zamknięto-otwarte (*ang. feedforward- feedback*) - Podrozdział 5.3 oraz proporcjonalno-całkowe (*ang. proportional integral*) - Podrozdział 5.4.

Przeprowadzone badania pokazały, że każda z tych metod umożliwia uzyskanie zadanego sygnału wyjściowego, jednak metody sterowania bezpośredniego i pośredniego nie zapewniają usunięcia wpływu zakłóceń. W przypadku sterowania zamknięto-otwartego wszystkie trzy cele są zapewnione, dodatkowo istnieje możliwość zastosowania metody stabilizacji opartej na dopasowaniu modelu, jednak poważną wadą tego podejścia jest fakt, że wymaga ono wyznaczenia wektorów stanu i sterowania w stanie ustalonym. Może to negatywnie wpłynąć na działanie tej metody. Najbardziej skuteczną (z pkt. widzenia spełnienia wymaganych celów) wydaje się metoda oparta na sterowaniu proporcjonalno-całkowym, w której nie można zaobserwować żadnej wad poprzednich metod. Jako pewną niedogodność można tu jednak traktować fakt, iż przy tym podejściu rozmiary warunków LMI stosowanych przy syntezie wzrastają 2-krotnie (co negatywnie odbija się na szybkości obliczania macierzy sterowników). Oczywistym jest, że ten znaczenie tego ograniczenia może być zmniejszone przez zastosowanie jednej z metod zaprezentowanych w Rozdziale 4 niniejszej rozprawy.

Nadmienić tu należy, że wyniki zaprezentowane w Rozdziale 5 są oryginalnymi osiągnięciami autora i zostały (bądź wkrótce zostaną) opublikowane na forum krajowym i międzynarodowym.

Podsumowanie

Tematyka niniejszej rozprawy odnosi się do opracowania i sprawdzenia numerycznie efektywnych warunków analizy i syntezy Liniowych Procesów Powtarzalnych. Takie umotywowanie dziedziny badań zostało podyktowane faktem, że istniejące techniki umożliwiające rozwiązywanie rozważanych zadań, z praktycznego pkt. widzenia, nie mogą być stosowane. Niska stosowalność dostępnych warunków, w zależności od rozważanego zadania wynika z poniższych faktów:

- część rozważanych warunków analizy należy do klasy zadań $\mathcal{NP}$-trudnych,
- wykorzystanie pozostałych rozważanych warunków wymaga przetwarzaniu macierzy o potencjalnie ogromnych rozmiarach, co wpływa negatywnie na szybkość obliczeń i może powodować powstawanie błędów numerycznych.

Dodatkowym ograniczeniem w stosowaniu istniejących warunków jest to, że przedstawiają one jedynie rozwiązanie zadania analizy. Nie dostarczają one możliwości naturalnego przekształcenia do warunków syntezy.

W celu wyeliminowania, bądź częściowego ograniczenia powyżej wymienionych problemów, w pracy zaprezentowano szereg wyników, które, dzięki zastosowaniu Liniowych Nierówności Macierzowych, okazały się skutecznym sposobem prowadzącym do:

- znacznego przyspieszenia otrzymywania rozwiązania (wynika to z faktu, że zadania LMI są, w gruncie rzeczy, traktowane jako zadania klasy $\mathcal{P}$ i rozwiązywane jako zadania optymalizacji wypukłej; dla tak sformułowanych zadań rozwiązanie jest otrzymywane w czasie wielomianowym),
- możliwości przedstawienia oryginalnych zadań $\mathcal{NP}$-trudnych w postaci przybliżonych zadań z klasy $\mathcal{P}$,
- ograniczenia możliwości wystąpienia błędów numerycznych, dzięki przetwarzaniu macierzy symetrycznych, dodatnio określonych.

Pomimo zastosowania LMI do rozwiązywania zadań analizy i syntezy procesów powtarzalnych, w dalszym ciągu istnieją pewne ograniczenia dotyczące ich stosowania. Szczególnie odnosi się to rozmiaru rozwiązywanego problemu. A zatem problem można teraz przedstawić następująco: dla otrzymanego warunku LMI, operującego na ogromnych macierzach, opracować efektywne sposoby, umożliwiające rozwiązanie postawionego zadania. W niniejszej rozprawie przedstawiono trzy takie rozwiązania, tj.

- rozwiązanie „sprzętowe", polegające na zapewnieniu dużej mocy obliczeniowej poprzez wykorzystanie technik obliczeń równoległych (klastry),
- opracowanie podejścia iteracyjnego, umożliwiającego wykorzystanie zalet procedur iteracyjnych – stopniowe zbliżanie się do rozwiązania (algorytm sukcesywnej stabilizacji),
- zaproponowanie metody uproszczenia spektrum rozważanej macierzy o dużych rozmiarach, poprzez wprowadzenia etapu wstępnej syntezy (preprocessing), a następnie rozwiązanie przetworzonego zadania syntezy (odseparowanie dynamik).

Jak widać, zaprezentowane metody mają ścisłe związki z informatyką.

Jako ostatni, jednak o szczególnej wadze, aspekt informatyczny pracy może być traktowany związek, jaki istnieje pomiędzy właściwościami dynamicznymi procesów powtarzalnych, a iteracyjnymi procedurami obliczeniowymi. We wstępie rozprawy opisano związki rozważanej klasy zadań analizy i syntezy z własnościami algorytmów iteracyjnych. Jak wspomniano, jedną ze wspomnianych praktycznych aplikacji, które z powodzeniem mogą być modelowane jako LRP, są zadania dotyczące iteracyjnego sterowania z uczeniem (ILC), które łatwo mogą być rozszerzone do modelowanie innych itereacyjnych procesów obliczeniowych, np. [30]. Zatem zapewnienie stabilności procesu powtarzalnego, odnosi się do zapewnienia zbieżności procesu iteracyjnego. Co więcej, przedstawione w rozprawie schematy syntezy zapewniające oprócz stabilności, dodatkowe własności układu w zamkniętej pętli sprzężenia zwrotnego (takie jak marginesy stabilności lub osiągnięcie żądanej postaci modelu), w odniesieniu do ILC, odnoszą się do „polepszenia" zachowania procedury iteracyjnej (np. przyspieszenia procedury minimalizacji błędu śledzenia). Można więc zauważyć, że istnieją silne powiązania pomiędzy własnościami Liniowych Procesów Powtarzalnych (teoria systemów), a własnościami procedury iteracyjnej (informatyka).

Reasumując, zagadnienia rozważane w poniższej rozprawie odnoszą się do komputerowo wspomaganego rozwiązywania zadań analizy i syntezy Liniowych Procesów Powtarzalnych. Otrzymane wyniki pokazują, że dzięki zastosowaniu podejść z szeroko pojętej dziedziny Informatyki, możliwe jest efektywne rozwiązywanie postawionych problemów. Osiągnięcia zaprezentowane w niniejszej rozprawie obejmują zarówno aspekty teoretyczne jak i praktyczne/aplikacyjne. Uzyskane wyniki udowadniają przyjętą tezę.

Bibliography

[1] N. K. Bose, *Applied multidimensional systems theory.* Van Nostrand-Reinhold, 1982.

[2] N. K. Bose, ed., *Multidimensional Systems Theory. Progress, Directions and Open Problems in Multidimensional Systems.* Dordrecht, The Netherlands: D. Reidel Publishing Company, 1985.

[3] T. Kaczorek, *Two-dimensional Linear Systems*, vol. 68 of *Lecture Notes in Control and Information Sciences.* Springer-Verlag, Berlin, 1985.

[4] E. Zerz, *Topics in Multidimensional Linear Systems Theory*, vol. 256 of *Lecture Notes in Control and Information Sciences.* London, England: Springer-Verlag, 2000.

[5] K. Gałkowski and J. Wood, eds., *Multidimensional Signals, Circuits and Systems.* Systems and Control Book Series, London, England: Taylor and Francis, 2001.

[6] K. Gałkowski, B. Sulikowski, and W. Paszke, "Układy wielowymiarowe," *Pomiary Automatyka Kontrola*, vol. 2–3, pp. 53–58, 2003. special issue, in Polish.

[7] C. Du and L. Xie, *H_∞ Control and Filtering of Two-dimensional Systems*, vol. 278 of *Lecture Notes in Control and Information Sciences.* Springer-Verlag, 2002.

[8] W.-S. Lu and A. Antoniou, *Two-dimensional Digital Filters*, vol. 80 of *Electrical Engineering and Elecronics.* New York, USA: Marcel Dekker, Inc, 1992.

[9] S. Basu, "Multidimensional causal, stable, perfect reconstruction filter banks," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 832–842, 2002.

[10] Y. Q. Shi and X. M. Zhang, "A new two-dimensional interleaving technique using successive packing," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 779–789, 2002.

[11] R. Rabenstein and P. Steffen, "Stability of image sequence processing systems," *AEU, Band 37*, no. 7/8, pp. 261–266, 1983.

[12] R. N. Bracewell, *Two-dimensional Imaging.* Prentice Hall Signal Processing Series, Upper Saddle River, USA: Prentice Hall Inc., 1995.

[13] D. E. Dudgeon and R. M. Merserau, *Multidimensional Digital Signal Processing.* Prentice-Hall Signal Processing Series, Englewood Cliffs, USA: Prentice Hall, 1984.

[14] A. Handkiewicz, M. Kropidłowski, and M. Łukowiak, "Two-dimensional signal processing based on FPGA," in *Proc. 2nd Int. Workshop on Multidimensional (nD) Systems*, (Czocha Castle, Lower Silesia, Poland), pp. 143–148, 27 - 30 June 2000.

[15] R. Rabenstein, P. Steffen, and S. Spors, "A tutorial on the representation of two-dimensional wave fields by multidimensional signals," in *The Fourth International Workshop on Mutlidimensional Systems - NDS 2005*, (Wuppertal, Germany), pp. 90–97, 2005. CDROM.

[16] J. Velten, M. Krips, and A. Kummert, "Realization of nD signal processing tasks in high-speed applications," in *The Fourth International Workshop on Mutlidimensional Systems - NDS 2005*, (Wuppertal, Germany), pp. 24–30, 2005. CDROM.

[17] E. Rogers and D. H. Owens, *Stability Analysis for Linear Repetitive Processes*, vol. 175 of *Lecture Notes in Control and Information Sciences*. Berlin, Germany: Springer-Verlag, 1992.

[18] S. E. Benton, *Analysis and Control of Linear Repetitive Processes*. University of Southampton, UK.: PhD Dissertation, 2000.

[19] E. Rogers and D. H. Owens, "Two decades of research on linear repetitive processes," in *Mutidimensional Signals, Circuits and Systems* (K. Gałkowski and J. Wood, eds.), Systems and Control Book Series, pp. 107–147, London, England: Taylor and Francis, 2001.

[20] E. Rogers, K. Gałkowski, and D. H. Owens, *Control Systems Theory and Applications for Linear Repetitive Processes*. Lecture Notes in Control and Information Sciences, Springer-Verlag, 2005. in print.

[21] R. P. Roesser, "A discrete state-space model for linear image processing," *IEEE Transactions on Automatic Control*, vol. 20, no. 1, pp. 1–10, 1975.

[22] E. Fornasini and G. Marchesini, "Doubly-indexed dynamical systems: state-space models and structural properties," *Mathematical Systems Theory*, vol. 12, pp. 59–72, 1978.

[23] S. Foda and P. Agathoklis, "Control of the metal rolling process: a multidimensional system approach," *Journal of the Franklin Institute*, vol. 329, no. 2, pp. 317–332, 1992.

[24] M. Yamada and O. Saito, "2D model-following servo system," *Multidimensional Systems and Signal Processing*, vol. 10, pp. 71–91, 1996.

[25] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Control and disturbance rejection for discrete linear repetitive processes," *Multidimensional Systems and Signal Processing*, vol. 16, no. 2, pp. 199–216, 2005.

[26] N. Amann, *Optimal Algorithms for Iterative Learning Control*. PhD thesis, University of Exeter, UK, 1996.

[27] N. Amann, D. H. Owens, and E. Rogers, "Predictive optimal iterative learning control," *International Journal of Control*, vol. 69, no. 2, pp. 203–226, 1998.

[28] R. W. Longman, "Iterative learning control - dynamic systems that learn in time and repetitions," in *Proc. 2nd Int. Workshop on Multidimensional (nD) Systems*, (Czocha Castle, Lower Silesia, Poland), pp. 55–65, 27-30 June 2000.

[29] D. H. Owens, N. Amann, E. Rogers, and M. French, "Analysis of linear iterative learning control schemes - a 2D systems/repetitive processes approach," *Multidimensional Systems and Signal Processing*, vol. 11, no. 1-2, pp. 125–177, 2000.

[30] P. D. Roberts, "Numerical investigations of a stability theorem arising from 2-dimensional analysis of an iterative optimal control algorithm," *Multidimensional Systems and Signal Processing*, vol. 11, no. 1-2, pp. 109–124, 2000.

[31] P. D. Roberts, "Two-dimensional analysis of an iterative nonlinear optimal control algorithm," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 872–878, 2002.

[32] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 9, pp. 1478–1495, 2003.

[33] J. E. Kurek and M. B. Zaremba, "Iterative learnig control synthesis based on 2-D system theory," *IEEE Transactions on Automatic Control*, vol. 38, no. 1, pp. 121–125, 1993.

[34] Y. Chen and C. Wen, *Iterative Learning Control. Convergence, Robustness and Applications*, vol. 248 of *Lecture Notes in Control and Information Sciences.* London, England: Springer-Verlag, 1999.

[35] P. D. Roberts, "Numerical investigation of a stability theorem arising from 2-dimensional analysis of an iterative optimal control algorithm," *Multidimensional Systems and Signal Processing*, vol. 11, no. 1/2, pp. 109–124, 2000.

[36] K. L. Moore, *Iterative Learning Control for Deterministic Systems.* Advances in Indrustrial Control Series, London, England: Springer-Verlag, 1983.

[37] A. D. Poularikas and S. Seely, *Signals and Systems.* PWS Engineering, Boston, 1985.

[38] T. Kaczorek, *Teoria Sterowania i Systemów.* Wydawnictwo Naukowe PWN, Warszawa, wyd. 2 poprawione ed., 1996. in Polish.

[39] D. H. Owens, E. Rogers, and K. Gałkowski, "Control theory and applications for repetitive processes," in *Advances in control: highlights of ECC '99*, pp. 327–333, London: Springer, 1999.

[40] J. Gramacki, *Stability and stabilisation linear repetitive processes.* PhD thesis, Technical University of Zielona Góra, 1999. in Polish.

[41] O. Toker and H. Özbay, "On the complexity of purely complex computation μ and related problems in multidimensional systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 409–414, 1998.

[42] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities In System And Control Theory*, vol. 15 of *SIAM studies in applied mathematics.* SIAM, Philadelphia, 1994.

[43] P. Gahinet and P. Apkarian, "A Linear Matrix Inequality approach to H_∞ control," *International Journal Of Robust and Nonlinear Control*, vol. 4, pp. 421–448, 1994.

[44] P. Gahinet, A. Nemirowski, A. J. Laub, and M. Chilali, *LMI Control Toolbox for use with MATLAB*. The Mathworks Partner Series, The MathWorks Inc., 1995.

[45] M. Chilali and P. Gahinet, "H_∞ design with pole placement constraints: an LMI approach," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 358–367, 1996.

[46] K. Gałkowski, W. Paszke, B. Sulikowski, E. Rogers, and D. Owens, "LMI based stability analysis and controller design for a class of 2D continuous-discrete linear systems," *American Control Conference, Anchorage, USA*, 2002. CD-ROM.

[47] K. Gałkowski, E. Rogers, S. Xu, J. Lam, and D. H. Owens, "LMIs - a fundamental tool in analysis and controller design for discrete linear repetitive processes," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 768–778, 2002.

[48] K. Gałkowski, J. Lam, E. Rogers, S. Xu, B. Sulikowski, W. Paszke, and D. H. Owens, "LMI based stability analysis and robust controller design for discrete linear repetitive processes," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 13, pp. 1195–1211, 2003.

[49] K. Gałkowski, W. Paszke, E. Rogers, S. Xu, J. Lam, and D. H. Owens, "Stability and control of differential linear repetitive processes using an LMI setting," *IEEE Transactions on Circuits and Systems - part II- Analog and Digital Signal Processing*, vol. 50, no. 10, pp. 662–666, 2003.

[50] W. Paszke, *Analisys and Synthesis of multidimensional systems classes using Linear Matrix Inequality methods*. University of Zielona Gora, Poland: PhD Dissertation, 2005.

[51] L. Xu, Q. Wu, Z. Lin, and Y. Anazawa, "A μ approach to robust stability analysis of nd discrete systems," *Multidimensional Systems and Signal Processing*, vol. 15, pp. 277–293, 2004.

[52] P.-A. Bliman, "Lyapunov equation for the stability of 2-D systems," *Multidimensional Systems and Signal Processing*, vol. 13, pp. 201—222, 2002.

[53] K. Gałkowski, B. Sulikowski, E. Rogers, and D. H. Owens, "Generalized discrete linear repetitive processes," in *Methods and Models in Automation and Robotics - MMAR 2001: Proceedings 7th IEEE International Conference*, vol. 1, (Miedzyzdroje, Poland), pp. 119–123, 2001. CDROM.

[54] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Successive stabilization of a class of 2D systems," in *Mathematical Theory of Networks and Systems - MTNS 2002*, (South Bend, USA), 2002. CDROM.

[55] K. Gałkowski, B. Sulikowski, W. Paszke, E. Rogers, and D. H. Owens, "Stability and controllability properties of extended discrete linear repetitive processes," in *Materiały Krajowej Konferencji Robotyki, Konferencje: 47*, no. 103/2 in Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej, (Lądek Zdrój, Poland), pp. 5–14, 2001.

[56] W. Paszke, B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Numeryczna analiza pewnych klas liniowych procesów powtarzalnych," in *Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim: III Krajowa Konferencja*, vol. 1, (Kraków, Poland), pp. 375–380, 2001. CDROM, in Polish.

[57] K. Gałkowski, E. Rogers, and D. H. Owens, "On the structure of singular discrete linear repetitive processes," in *Proc. IFAC/IEEE Symposium on Systems Structure and Control*, (Prague, Czech Republic), August 29-31, 2001 2001. CD-ROM.

[58] K. Gałkowski, B. Sulikowski, E. Rogers, and D. H. Owens, "Analiza osobliwych dyskretno-ciagłych liniowych procesów powtarzalnych z zastosowaniem LMI," in *XIV Krajowa Konferencja Automatyki - KKA 2002*, vol. 1, (Zielona Góra, Poland), pp. 283–288, 2002. CDROM, in Polish.

[59] K. Gałkowski, E. Rogers, and D. H. Owens, "Analysis of linear repetitive processes with dynamic boundary conditions.," tech. rep., Dept. Electronics and Computer Science,University of Southampton, UK, 1998.

[60] S. Xu, J. Lam, Z. Lin, K. Gałkowski, W. Paszke, B. Sulikowski, E. Rogers, and D. H. Owens, "Positive real control of two-dimensional systems: Roesser models and linear repetitive processes," *International Journal of Control*, vol. 76, no. 11, pp. 1047–1058, 2003.

[61] K. Gałkowski, E. Rogers, J. Wood, S. E. Benton, and D. H. Owens, "One-dimensional equivalent model and related approaches to the analysis of discrete nonunit memory linear repetitive processes„" *Circuits, Systems and Signal Processing*, vol. 21, no. 6, pp. 525–534, 2002.

[62] A. Gramacki, *Discretisation methods of linear differential repetitive processes.* PhD thesis, Technical University of Zielona Góra, 1999. in Polish.

[63] T. Kaczorek, "Generalized 2D continuous-discrete linear systems with delay," *Applied Mathematics and Computer Science*, vol. 5, no. 3, pp. 439–454, 1995.

[64] K. Gałkowski, *State-Space Realisations of Linear 2-D Systems with Extensions to the General nD Case*, vol. 263 of *Lecture Notes in Control and Information Sciences.* Springer Verlag, London, 2001.

[65] J. Kurek, "The general state–space model for two–dimensional linear digital systems," *IEEE Transactions on Automatic Control*, vol. AC-30, pp. 600–602, June 1985.

[66] L. Hładowski, K. Gałkowski, and E. Rogers, "Repetitive processes - a way towards modelling hybrid systems," in *Computer Information Systems and Industrial Management Applications - CISIM '03*, (Ełk, Poland), pp. 125–132, Wyzsza Szkola Finansow i Zarzadzania w Bialymstoku, Białystok, 2003.

[67] D. H. Owens and E. Rogers, "Stability analysis for a class of 2D continuous-discrete linear systems with dynamic boundary conditions," *Systems and Control Letters*, vol. 37, pp. 55–60, 1999.

[68] K. Gałkowski, E. Rogers, A. Gramacki, J. Gramacki, and D. H. Owens, "Controlling of dynamics of discrete linear repetitive processes by rejecting the effects of dynamic pass initial conditions," *The First International Workshop on Multidimensional (nD) Systems NDS–98, Łagów, Zielona Góra, Poland*, pp. 63–68, 1998.

[69] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[70] K. Gałkowski, B. Sulikowski, E. Rogers, and D. H. Owens, "Successive stabilization of highly dimensional repetitive processes," in *Digital Signal Processing and its Applications - DSPA 2003*, (Moscow, Russia), 2003. CDROM.

[71] K. Gałkowski, E. Rogers, and D. H. Owens, "Matrix rank based conditions for reachability / controllability of discrete linear repetitive processes," *Linear Algebra and its Applications*, vol. 275–276, pp. 201–224, 1998.

[72] K. Gałkowski, E. Rogers, and D. Owens, "New 2D models and a transition matrix for discrete linear repetitive processes," *International Journal of Control*, vol. 72, no. 15, pp. 1365–1380, 1999.

[73] E. Rogers, A. Gramacki, J. Gramacki, D. H. Owens, and K. Gałkowski, "Higher order discretisation methods for a class of 2-D continuous-discrete linear systems," *IEE Proceedings - Circuits, Devices and Systems*, vol. 146, no. 6, pp. 315–320, 1999.

[74] K. Gałkowski, "Higher order discretization of 2-D systems," *IEEE Transactions on Circuits and Systems - I : Fundamental Theory and Applications*, vol. 47, no. 5, pp. 713–722, 2000.

[75] A. Gramacki, J. Gramacki, K. Gałkowski, E. Rogers, and D. H. Owens, "From continuous to discrete models of linear repetitive processes," *Archives of Control Sciences*, vol. 12, no. 1-2, pp. 151–185, 2002.

[76] J. Bochniak and K. Gałkowski, "LMI-based analysis for continuous-discrete linear shift-invariant nD systems," *Journal of Circuits, Systems, and Computers*, vol. 14, no. 2, pp. 307–332, 2005.

[77] K. L. Moore, "A non-standard iterative learning control approach to tracking periodic signals in discrete-time non-linear systems," *International Journal of Control*, vol. 73, no. 10, pp. 955–967, 2000.

[78] D. H. Owens and J. Hatonen, "Parameter optimal iterative learning control: a unifying formulation and analisys," in *The Fourth International Workshop on Mutlidimensional Systems - NDS 2005*, (Wuppertal, Germany), pp. 129–135, 2005. CDROM.

[79] R. Merry, R. van de Molengraft, and M. Steinbuch, "Disturbances and model uncertainties in iterative learning control," in *The Fourth International Workshop on Mutlidimensional Systems - NDS 2005*, (Wuppertal, Germany), pp. 136–141, 2005. CDROM.

[80] O. Bachelier, J. Bernussou, M. C. de Oliveria, and J. C. Geromel, "Parameter dependent Lyapunov control design : numerical evaluation," *Proceedings of 38th IEEE International Conference on Decision and Control*, pp. 293–297, 1999.

[81] C. A. R. Crusius and A. Trofino, "Sufficient LMI conditions for output feedback control problems," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 1053–1057, 1999.

[82] M. Safonov, K. C. Goh, and J. H. Ly, "Control System Synthesis via Bilinear Matrix Inequalities," in *Proceedings of the American Control Conference (ACC)*, vol. 1, (Baltimore, USA), pp. 45–49, June 29 - July 1 1994.

[83] O. Toker and H. Özbay, "On the NP-hardness of solving Bilinear Matrix Inequalities and simultaneous stabilization with static output feedback," in *Proceedings of the American Control Conference (ACC)*, vol. 4, (Seattle, USA), pp. 2525–2526, June 21-23 1995.

[84] L. El Ghaoui and S. I. Niculescu, eds., *Advances in Linear Matrix Inequality Methods in Control*. Advances in Design and Control, SIAM, 1999.

[85] D. Henrion and M. Kocvara, *LMI and BMI optimization: algorithms and applications in control,*. Prague, Czech Republic, February 2005. Available at http://www.laas.fr/ henrion/courses/lmi05/.

[86] D. Henrion, M. Kocvara, and M. Stingl, "Solving simultaneous stabilization BMI problems with PENNON," Research Report 03549, LAAS-CNRS, LAAS, Toulouse, France, 2003.

[87] C. Scherer and S. Weiland, *Linear Matrix Inequalities in Control*. Delft University of Technology, Delft, The Netherlands, 2002.
Available at http://www.cs.ele.tue.nl/SWeiland/lmi.html.

[88] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Massachusetts, USA, 2004.

[89] K. Fujisawa, M. Kojima, and K. Nakata, "SDPA (Semidefinite Programming Algorithm) - User's Manual," Technical Report B-308, Tokyo Institute of Technology, 2000.
Available at http://rnc.r.dendai.ac.jp/∼fujisawa/sdpa_doc.pdf.

[90] B. Mohar and S. Poljak, "Eigenvalues in combinatorial optimization," in *Combinatorial and graph-theoretic problems in linear algebra* (R. Brualdi, S. Friedland, and V. Klee, eds.), vol. 50 of *IMA Volumes in Mathematics and Its Applications*, Springer-Verlag, 1993.

[91] S. Delorme and S. Poljak, "Laplacian eigenvalues and the maximum cut problem," *Mathematical Programming*, vol. 62, pp. 557–574, 1993.

[92] R. B. Boppana, "Eigenvalues and graph bisection: an average case analysis," in *Proceedings of the 28th Annual Symposium on Computer Science*, pp. 280–285, 1987.

[93] L. Lovasz, "On the Shannon capacity of the graph," *IEEE Transactions on Information Theory*, vol. 25, pp. 1–7, 1979.

[94] M. L. Overton, "Large-scale optimization of eigenvalues," *SIAM Journal of Optimization*, vol. 2, pp. 88–120, 1992.

[95] F. Jarre, "An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices," *SIAM Journal of Control and Optimization*, vol. 31, pp. 1360–1377, 1993.

[96] S. J. Benson, "Parallel computing on semidefinite programs," Tech. Rep. ANL/MCS-P939-0302, Mathematics and Computer Science Division, Argonne National Laboratory, 2003. Available at http://www-unix.mcs.anl.gov/ benson/dsdp/.

[97] M. Yamashita, K. Fujisawa, and M. Kojima, "SDPARA: Semidefinite programming algorithm parallel version," *Parallel Computing*, vol. 29, pp. 1053–1067, 2003.

[98] K. Nakata, M. Yamashita, K. Fujisawa, and M. Kojima, "A parallel primal-dual interior point method for semidefinite programs using positive definite matrix completion," Research report B-398, Department of Mathematics and Computer Sciences, Tokyo Institute of Technology, Tokyo, Japan, 2003.
Available at http://www.is.titech.ac.jp/∼kojima/sdpara/index.html.

[99] L. Khachiyan, "A polynomial algorithm in linear programming," *Soviet Mathematics Doklady*, vol. 20, pp. 191–194, 1979.

[100] Y. Nesterov and A. Nemirovskii, *Interior-point Polynomial Algorithms in Convex Programing: Theory and Applications*, vol. 13 of *SIAM Studies in Applied and Numerical Mathematics*. Philadelphia, USA: SIAM, 1994.

[101] L. Vandenberghe and V. Balakrishnan, "Algorithms and software for LMI problems in control," *IEEE Control System Magazine*, vol. 17, no. 5, pp. 89–95, 1997.

[102] G. Meinsma, *Interior Point Methods - mini course*. The University of Twente, Enschede, The Netherlands, 1997.
Available at http://wwwhome.math.utwente.nl/∼meinsmag/courses/ipm.html.

[103] P. Gahinet and A. Nemirovski, "The Projective Method for solving linear matrix inequalities," *Mathematical Programming*, vol. 77, pp. 163–190, 1997.

[104] A. Nemirovskii and P. Gahinet, "The projective method for solving linear matrix inequalities," in *Proceedings of the American Control Conference (ACC)*, (Baltimore, USA), pp. 840–844, June 29-July 1 1994.

[105] S. J. Benson, Y. Ye, and X. Zhang, "Solving large-scale sparse semidefinite programs for combinatorial optimization," *SIAM Journal on Optimization*, vol. 10, pp. 443–461, 2000.

[106] M. Kocvara, F. Leibfritz, M. Stingl, and D. Henrion, "A nonlinear SDP algorithm for static output feedback problems in COMPlib," Research Report 04508, LAAS-CNRS, LAAS, Toulouse, France, 2004.

[107] K. Fujisawa, M. Kojima, and K. Nakata, "Exploiting the sparsity in primal-dual interior-point methods for semidefinite programming," *Mathematical Programming*, vol. 79, pp. 235–253, 1997.

[108] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, "Exploiting the sparsity in semidefinite programming via matrix completion II: Implementation and numerical result," *Mathematical Programming*, vol. 95, pp. 303–327, 2003.

[109] The Mathworks Inc., *Matlab. The Language of Technical Computing*. Natick, USA, 2004.
Available at http://www.mathworks.com/access/helpdesk/help/helpdesk.html.

[110] C. Gomez, *Engineering and Scientific Computing with Scilab.* Boston, USA: Birkhäuser, 1999.

[111] K. C. Toh, R. H. Tutuncu, and M. J. Todd, *SDPT3 version 3.02 – a MATLAB software for semidefinite-quadratic-linear programming.* Department of Mathematics, National University of Singapore, Singapore, 2002.
Available at http://www.math.nus.edu.sg/∼mattohkc/sdpt3.html.

[112] M. Kočvara and M. Stingl, "PENNON - A Code for Convex Nonlinear and Semidefinite Programming," *Optimization Methods and Software*, vol. 18, no. 3, pp. 317–330, 2003.

[113] S.-P. Wu, L. Vandenberghe, and S. Boyd, *MAXDET manual.* Stanford University, 1996. Available at http://www.stanford.edu/∼boyd/MAXDET.html.

[114] S.-P. Wu and S. Boyd, *Recent Advances in LMI Methods for Control*, ch. SDPSOL: A parser/solver for semidefinite programs with matrix structure, pp. 79–91. Advances in Design and Control, SIAM, 2000.

[115] S. Benson and Y. Ye, "DSDP4 - A Software Package Implementing the Dual-Scaling Algorithm for Semidefinite Programming," Tech. Rep. ANL/MCS-TM-255, Mathematics and Computer Science Division, Argonne National Laboratory, 2002.

[116] E. Andersen, *MOSEK*, 2001. Available at http://www.mosek.com/.

[117] F. Xhafa and G. Navarro, "A Maple package for Semidefinite Programming," Research report LSI-96-28-R, Dept. Llenguatges i Sistemes Informatics - Universitat Politènica de Catalunya, 1996.

[118] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11-12, pp. 625–653, 1999. Special issue on Interior Point Methods (CD supplement with software).

[119] K. Fujisawa, Y. Futakata, M. Kojima, S. Matsuyama, S. Nakamura, K. Nakata, and M. Yamashita, "SDPA-M (semidefinite Programming Algorithm in MATLAB), user's manual – version 6.2.0," Technical Report B-359, Tokyo Institute of Technology, 2000. Available at http://rnc.r.dendai.ac.jp/∼fujisawa/sdpam_doc.pdf.

[120] K. Fujisawa, M. Fukuda, M. Kojima, K. Nakata, and M. Yamashita, "SDPA-C (semidefinite Programming Algorithm – completion method), user's manual – version 6.10," Technical Report B-409, Tokyo Institute of Technology, 2004.
Available at http://rnc.r.dendai.ac.jp/∼fujisawa/sdpac_doc.pdf.

[121] R. Nikoukhah, F. Delebecque, and L. E. Ghaoui, *LMITOOL: a Package for LMI Optimization in Scilab.* INRIA, France.
Available at ftp://ftp.inria.fr/INRIA/Scilab/documentation/pdf/lmi.pdf.

[122] H. Mittelmann, "An independent benchmarking of SDP and SOCP solvers," *Mathematical Programming*, vol. 95, pp. 407–430, 2000.

[123] B. Borchers, "SDPLIB 1.2, a library of semidefinite programming test problems," *Optimization Methods and Software*, vol. 11-12, pp. 683–690, 1999.

[124] J. Löfberg, *YALMIP 3*, 2004. Available at http://control.ee.ethz.ch/∼joloef/yalmip.msql.

[125] L. Vandenberghe and S. Boyd, "Semidefinite programing," *SIAM review*, vol. 38, pp. 49–95, 1996.

[126] G. E. Karniadakis and R. M. Kirby, *Parallel Scientific Computing in C++ and MPIP: A seamless approach to parallel algorithms and their implementation*. Cambridge University Press, Cambridge, Massachusetts, USA, 2002.

[127] T. Sterling, J. Salmon, D. J. Becker, and D. Savarese, *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters*. MIT Press, USA, 1999.

[128] R. W. Hockney, *The Science of Computer Benchmarking*. Software, Environments, Tools, SIAM, San Diego, 1996.

[129] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI: The Complete Reference*. The MIT Press, Cambridge, Massachusetts, 1996.

[130] P. S. Pacheco, "A user's guide to MPI," Tech. Rep. CA 94117, Department of Mathematics, University of San Francisco, 1998.

[131] K. Gałkowski, W. Paszke, B. Sulikowski, E. Rogers, and D. H. Owens, "LMI based stability analysis and robust controller design for discrete linear repetitive processes," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 13, pp. 1195–1211, 2003.

[132] K. Gałkowski, W. Paszke, B. Sulikowski, E. Rogers, and D. H. Owens, "LMI based stability analysis and controller design for a class of 2D continuous-discrete linear systems," in *American Control Conference, ACC-2002*, (Anchorage, USA), pp. 29–34, 2002. CDROM.

[133] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Feedforward/feedback control of discrete linear repetitive processes with disturbances applied to a material rolling process," in *Proceedings of the 2003 IEEE International Conference on Control Applications - CCA 2003*, (Istanbul, Turkey), 2003. CDROM.

[134] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Decoupling and iterative approaches to the control of an applicable class of 2D linear systems," *European Journal of Control*, 2005. in review.

[135] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Output feedback control of discrete linear repetitive processes," *Automatica*, vol. 40, no. 12, pp. 2163–2173, 2004.

[136] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Further results on output feedback control of discrete linear repetitive processes," in *First International Conference in Informatics in Control, Automation and Robotics*, vol. 3, (Setubal, Portugal), pp. 264–269, 2004.

[137] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "LMI based output feedback control of discrete linear repetitive processes," in *American Control Conference - ACC 2004*, (Boston, USA), pp. 1998–2003, 2004. CDROM.

[138] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Proportional plus integral control and disturbance rejection for differential linear repetitive processes," in *American Control Conference - ACC 2005*, (Portland, USA), pp. 1782–1787, 2005. CDROM.

[139] K. Gałkowski, W. Paszke, B. Sulikowski, E. Rogers, and D. H. Owens, "Stability and control of a physical class of 2D continuous-discrete linear systems using an LMI setting," in *American Control Conference - ACC 2003*, vol. 6, (Denver, USA), pp. 5058–5063, 2003. CDROM.

[140] R. A. Hyde, *H-infinity Aerospace Control Design - A VSTOL Flight Application*. Advances in Industrial Control Series, Springer Verlag, 1995.

[141] X. Li, "Design and parametrization of 2D state observer," *Transaction of the Society of Instrument and Control Engineers (Academic Journal)*, vol. 8, no. 33, pp. 855–861, 1997.

[142] M. Bisiacco and M. E. Valcher, "Unknown input observers for 2-D systems," in *Mathematical Theory of Networks and Systems - MTNS 2002*, (Leuven, Belgium), 2004.

[143] M. Chilali, P. Gahinet, and P. Apkarian, "Robust pole placement in LMI regions," *IEEE Transactions on Automatic Control*, vol. 44, no. 12, pp. 2257–2270, 1999.

[144] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Control of discrete linear repetitive processes with application to a material rolling process," in *European Control Conference - ECC '03*, (Cambridge, England), 2003. CDROM.

[145] B. Sulikowski, K. Gałkowski, E. Rogers, D. H. Owens, and A. Kummert, "Output feedback/feedforward control of discrete linear repetitive processes for performance and disturbance rejection," in *Proceedings of 25th IEEE American Control Conference, ACC 2006*, 2006. submitted.

[146] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Proportional plus integral control for a class of 2D linear systems," in *Methods and Models in Automation and Robotics - MMAR 2004: Proceedings of the 10th IEEE International Conference*, vol. 1, (Miedzyzdroje, Poland), pp. 335–340, 2004. CDROM.

[147] B. Sulikowski, K. Gałkowski, E. Rogers, and D. H. Owens, "Two term control of discrete linear repetitive processes," *Automatica*, 2004/2005. in review.

[148] B. Sulikowski, K. Gałkowski, E. Rogers, D. H. Owens, and W. Paszke, "Proportional plus integral control and disturbance rejection for discrete linear repetitive processes," in *The Fourth International Workshop on Mutlidimensional Systems- NDS 2005*, (Wuppertal, Germany), pp. 154–159, 2005. CDROM.

[149] E. Rogers, K. Gałkowski, A. Gramacki, J. Gramacki, and D. H. Owens, "Stability and controllability of a class of 2D systems with dynamic boundary conditions," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 2, pp. 181–195, 2002.

[150] G. P. Liu, G. R. Duan, and R. Dixon, "Robust control with stable proportional-integral-plus controllers," in *Proceedings of European Control Conference, ECC 2001*, (Porto, Portugal), pp. 3201–3206, 2001.

[151] W. Paszke, K. Gałkowski, E. Rogers, and D. H. Owens, "H2 control of differential linear repetitive processes," in *Proceedings of the 16th IFAC World Congress in Prague*, (Prague, Czech Repuplic), 2005. CDROM.

[152] J. Lam, S. Xu, Y. Zou, Z. Lin, and K. Gałkowski, "Robust output feedback stabilization for two-dimensional continuous systems in Roesser form," *Applied Mathematics Letters*, vol. 17, pp. 1331–1341, 2004.

[153] B. Cichy, K. Gałkowski, E. Rogers, A. Kummert, and D. H. Owens, "Control of discrete linear repetitive processes with uncertain sampling period and application to a physical example," in *The Fourth International Workshop on Multidimensional Systems - NDS 2005*, (Wuppertal, Niemcy), pp. 148–153, Wuppertal, GMT Gesellschaft für Medizin und Technik, 2005. CD-ROM.

[154] H. Melkote, B. Cloke, and V. Agarwal, "Modeling and compensator design for self servo-writing in disk drives," *American Control Conference*, pp. 737–742, 2003.

[155] C. D. Meyer, *Matrix Computations and Applied Linear Algebra*. Philadelphia, USA, SIAM, 2000.

[156] D. Henrion, M. Sebek, and V. Kucera, "Robust pole placement for second-order systems: an LMI approach," *Proceedings of the IFAC Symposium on Robust Control Design*, 2003. LAAS-CNRS Research Report No. 02324, July 2002.