Analysis and synthesis of multidimensional system classes using linear matrix inequality methods

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

Lecture Notes in Control and Computer Science
Volume 8

Wojciech Paszke

Analysis and synthesis of multidimensional system classes using linear matrix inequality methods

University of Zielona Góra Press, Poland

2005

Wojciech Paszke
Institute of Control and Computation Engineering
University of Zielona Góra
ul. Podgórna 50
65-246 Zielona Góra, Poland
e-mail: `W.Paszke@issi.uz.zgora.pl`

Referees:

- Andrzej DZIELIŃSKI, Warsaw University of Technology
- Dariusz UCIŃSKI, University of Zielona Góra

The text of this book was prepared based on the author's Ph.D. dissertation

Partially supported by the State Committee for Scientific Research (KBN)
in Poland

ISBN 83-89712-81-4

Camera-ready copy prepared from the author's LaTeX2$_\varepsilon$ files.
Printed and bound by University of Zielona Góra Press, Poland.

CONTENTS

Acknowledgments

First of all, I would like to express my gratitude to my supervisor Prof. Krzysztof Gałkowski. Without his constructive comments, useful suggestions and wealth of ideas this dissertation would have never been completed.

I am also grateful to Prof. James Lam from University of Hong Kong and Prof. Eric Rogers from University of Southampton for their advice and support. Their encouragement, knowledge in the field of linear matrix inequalities, multidimensional systems and repetitive processes greatly contributed to the value of this work.

I also thank my friends Maciej Patan, Bartłomiej Sulikowski and Bartosz Kuczewski for many enlightening discussions.

I would also like to thank my parents, my brother and my sister, for their unconditional love and support. Finally yet importantly, I dedicate this work to my wonderful and beloved Rita, without whom nothing would have been possible.

Notation

Abbreviations

1-D	one-dimensional
2-D	two-dimensional
n-D	n-dimensional (multidimensional)
ARE	Algebraic Riccati Equation
BMI(s)	Bilinear Matrix Inequality(ies)
EVP	Eigenvalue Problem
FMM	Fornasini-Marchesini Model
GEVP	Generalized Eigenvalue Problem
ILC	Iterative Learning Control
IPM	Interior-Point Method
LMI(s)	Linear Matrix Inequality(ies)
LRP(s)	Linear Repetitive Process(es)
LSE	Linear System of Equations
RM	Roesser Model
SDP	Semi-Definite Program(ming)
SVD	Singular Value Decomposition

Notation

$\mathbb{R}^n$	n-dimensional real vector space
$\mathbb{R}^{n\times m}$	set of real $n \times m$ matrices
$\mathbb{N}$	set of natural numbers
$\mathbb{Z}$	set of integer numbers
$\boldsymbol{A} \succ\ (\succeq)\ 0$	positive (semi-)definite matrix
$\boldsymbol{A} \prec\ (\preceq)\ 0$	negative (semi-)definite matrix
$\boldsymbol{A}^T$	transpose of a matrix
$\boldsymbol{A}^{-1}$	inverse of a matrix
$\boldsymbol{A}^{-T}$	$(\boldsymbol{A}^T)^{-1}$
$\boldsymbol{I}$	identity matrix
$\mathbf{0}$	null matrix
$\mathrm{trace}(\boldsymbol{A})$	trace of a matrix
$\mathrm{rank}(\boldsymbol{A})$	rank of a matrix
$\mathrm{diag}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n)$	matrix with diagonal elements $(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n)$
$\det(\boldsymbol{A})$	determinant of a matrix
$\ker(\boldsymbol{A})$	null space of a matrix
$\mathrm{Im}(\boldsymbol{A})$	range space of a matrix
$\mathrm{Co}\,(,\,,\,)$	convex hull
$\mathrm{Re}(\cdot)$	real part of a complex number
$\lambda(\boldsymbol{A})$	set of eigenvalues of a matrix
$\lambda_{\max}(\boldsymbol{A})$, $\lambda_{\min}(\boldsymbol{A})$	maximum, minimum eigenvalue of a matrix $\boldsymbol{A}$
$\overline{\sigma}(\boldsymbol{A})$, $\underline{\sigma}(\boldsymbol{A})$	maximum, minimum singular value of a matrix $\boldsymbol{A}$
$\rho(\boldsymbol{A})$	spectral radius of a matrix $\boldsymbol{A}$
$\forall$	for all

Chapter 1

INTRODUCTION

In the past three decades, progress in technology has been accompanied by the development of diverse and complicated systems. Among them, are multidimensional (n-D) systems characterized by rational functions, or matrices of several independent variables which can represent different space co-ordinates or mixed time and space variables. This is a result of information propagation in more than one independent direction which is the essential difference from the classical, or one-dimensional (1-D) case, where information propagates only in one direction.

The interest in n-D systems (Bose, 1982, 1985, 2001; Gałkowski *et al.*, 2003e; Gałkowski and Wood, 2001; Kaczorek, 1985; Zerz, 2000) has been motivated predominantly by a wide variety of applications, arising in both theory and practical applications. Examples of such applications are multidimensional signal and image processing (Bracewell, 1995; Dudgeon and Merserau, 1984; Handkiwicz *et al.*, 2000; Mese and Vaidyanathan, 2002), n-D coding and decoding (Miri and Aplevich, 2000; Shi and Zhang, 2002) and n-D filtering techniques (Basu, 2002; Lu and Antoniou, 1992), which are frequently used in computer graphics and animation for object rendering. Recently, repetitive processes have been the most investigated class of n-D systems (Rogers *et al.*, 2005; Rogers and Owens, 1992, 2001), which are clearly two-dimensional (2-D) systems, with applications ranging from long-wall coal cutting and metal rolling (see, for example, (Foda and Agathoklis, 1992; Gałkowski *et al.*, 2003d; Rogers and Owens, 1992)) to iterative learning control (ILC) schemes (Amann *et al.*, 1998; Longman, 2000; Owens *et al.*, 2000) and iterative algorithms for solving nonlinear dynamic optimal control problems based on the maximum principle (Roberts, 2000a,b, 2002).

While studying linear n-D systems and linear repetitive processes (LRPs), many analysis and synthesis problems arise. Among them are fundamental questions related to stability and stabilisation, feedback control, robust control and performance measure that should be provided with efficient methods of their solution. Unfortunately, linear n-D systems and LRPs cannot be analyzed by direct extension of existing methods from 1-D systems theory, because such an approach ignores their inherent n-D structure and results in computationally hard problems. This makes most n-D system analysis and synthesis tests not computationally effective and sometimes not feasible.

In view of computer-aided analysis and synthesis, the efficiency of a method involves two main aspects: avoiding extensive storage use and keeping the computational complexity as low as possible. The first requirement implies that the storage should be proportional to the amount of data defining the system. Due to the fact that large amounts of memory are available in modern computer workstations, the first requirement may be neglected. Concerning the second requirement, we focus on some key concepts from the theory of complexity, highlighting their relevance to systems and control theory (Blondel and Tsitsiklis, 2000b; Fu and Luo, 1997; Vidyasagar, 1998). A large proportion of the control problems (especially in 1-D systems theory) is algorithmically solvable with polynomial time computability, i.e. the running time of an algorithm on any problem instance of the size n increases no faster than some polynomial function in n. This class of problems is so-called $\mathcal{P}$, and the problems which belong to such a class are considered efficiently solvable. An example of such a problem in 1-D control theory, known to be polynomial-time solvable, is the stability problem. It can be converted into computing of system matrix eigenvalues and verifying whether their moduli are all smaller than 1 (discrete system case) or that they are all in the open-left half-plane (continuous system case). Since these operations are always performed in polynomial time, the stability problem belongs to class $\mathcal{P}$. It is important to note that an alternative method is the Routh test, which can also be answered in polynomial time. This clearly means that the described stability tests of a 1-D system can be efficiently solved even for high-order systems, and hence it is suitable for inclusion in a software package.

The problem is assigned to the $\mathcal{NP}$ (nondeterministic polynomial time) class if it is verifiable (but not necessarily solvable) in polynomial time, i.e. we can verify, in polynomial time, whether a proposed solution is correct. Other problems are shown to be $\mathcal{NP}$-hard (Blondel and Tsitsiklis, 2000b; Vidyasagar and Blondel, 2001), meaning that although these problems may be algorithmically solvable, no polynomial time algorithm is possible, assuming the validity of a long-standing conjecture in computer science ($\mathcal{P} \neq \mathcal{NP}$), which is widely believed to be true. An example of the $\mathcal{NP}$-hard problem in 1-D system theory is a question related to output feedback stabilisation with constraints, namely determining whether there exists a matrix $\boldsymbol{K}$ satisfying $\boldsymbol{K}_{\text{low}} < \boldsymbol{K} < \boldsymbol{K}_{\text{high}}$ and such that $\boldsymbol{A} + \boldsymbol{BKC}$ is stable (Vidyasagar, 1998), where matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{K}_{\text{low}}$ and $\boldsymbol{K}_{\text{high}}$ are given. However, many $\mathcal{NP}$-hard problems can be routinely solved in practice, either exactly or approximately using various methods, but generally they are only limited to low-scale problems. This makes computer implementations possible. Some of control problems are shown to be undecidable, that is, they are not amenable to an algorithmic solution and frequently the approximate versions of these problems are also computationally hard. An example of an undecidable problem in control theory is the one related to the stability of linear time-varying systems (Blondel and Tsitsiklis, 2000a).

It is a natural question to ask if analysis and synthesis problems in n-D systems theory can be considered effectively or satisfactorily solved. Unfortunately, nowadays, most of such problems are assumed to be $\mathcal{NP}$-hard or even undecidable. This means that solving many of n-D system analysis and synthesis problems is a nontrivial task. To see this, consider, for example, the stability problem. When dealing with 2-D (n-D) systems, representations in terms of rational functions of several independent variables are frequently used as a foundation for a systems analysis (Bose, 1977, 1982; Hätönen and Ylinen, 2003; Youla and Gnavi, 1979) and allow us to formulate stability tests. However, they turn out to be undecidable because we have to check an infinite set of system poles, which obviously cannot be done in finite time. Indeed, zeros of a 2-D (n-D) system characteristic polynomial (i.e. 2-D (n-D) system poles) are not isolated as in the 1-D case and, in general, they cannot form a finite set. Therefore, to date no effective tests (with the complexity being a polynomial in decision variables) exist for deciding if a polynomial of several independent variables has no zeroes in the closed bidisk (n-disk). Furthermore, the problems with computation of n-D system poles are accompanied by difficulties in applying the pole placement technique, since there is no link between the pole placement and the dynamic response of the n-D system. This immediately makes the controller synthesis a difficult task.

It is not a surprise that computational problems of the same kind occur when dealing with LRPs and we take account of the fact that they reveal an inherent 2-D system structure. As an example, the stability test for discrete LRPs can be considered. It is shown in (Rogers and Owens, 1992) that the standard test for stability along the pass, involves the computation of the eigenvalues of a potentially large matrix for all points on the unit circle in complex plane. This is clearly impossible in practice and means that the stability problem of LRPs is undecidable. In view of these computational difficulties, it is only possible to provide sufficient stability conditions, i.e. conditions which involve only computations for a finite number of points. On the other hand, it is difficult to specify the points for checking. However, in the case of LRP, information propagation in one of two separate directions only occurs over a finite duration and therefore the 1-D model of underlying dynamics can be provided. One of the implications of this fact, is the possibility of applying 1-D system analysis and synthesis methods. Nevertheless, the resulting process matrices depend on the pass length, which often makes 1-D system theory tests computationally ineffective (Gramacki, 1999b).

Another set of problems with the n-D system theory are those that are caused by the complexity of the underlying ring structure, i.e. polynomials in two and more indeterminates where the underlying ring does not have a division algorithm (Bose, 1977; Gałkowski, 2001a). The existence of a division algorithm for Euclidean ring forms constitutes a foundation for the algorithmic derivation of many canonical forms and solution techniques in 1-D system theory.

To overcome some of these problems, much effort has been dedicated to establishing mathematical tools, outside these required in 1-D system theory, which can be used to constitute computationally effective methods for n-D systems analysis and synthesis. Among recently developed methods are those based on behavioural theory (Pillai *et al.*, 2002; Wood *et al.*, 2001) and Groebner-basis (Basu, 2002; Curtin and Saba, 1999; Kalker *et al.*, 1995).

The behavioural approach (Polderman and Willems, 1997; Willems, 1991) to linear systems looks at system equations as just one possible way of representing a ‘behaviour’, that is a set of time trajectories of a vector of selected variables. This point of view allows us to consider some hard problems in n-D system theory, e.g.

n-D systems pole placement, but this requires wide-spread knowledge of abstract algebra (Shankar and Willems, 2000; Wood *et al.*, 2001). In addition to this, the behavioural approach is still under development and today, this approach cannot be widely applied in practice as many theoretical problems to overcome remain.

The interest in the Groebner bases, which have been introduced by Buchberger (Buchberger, 1985), has been motivated by developing algebraic methods and software to support them. Nevertheless, Groebner bases establish an automatic proving method for many non-trivial theorems which have immediate applications in n-D system theory (Lin, 2001; Lin *et al.*, 2001). They need to find an appropriate multivariate polynomial set associated with the problem at hand that is a hard task—worst-case complexity is doubly exponential in the number of variables (Bose, 1985). For this reason available software allows us to apply this method only to very small problems (systems with five or six variables), and therefore it is not very useful while solving engineering problems.

Recently, the most popular technique for formulating the stability test for n-D systems is that based on constructing Lyapunov functionals (Bliman, 2002; Hinamoto, 1997; Lu, 1994; Lu and Lee, 1985). Since an appropriate Lyapunov functional candidate is provided, the corresponding stability condition can be expressed in terms of bilinear matrix inequalities (BMIs) (Safonov *et al.*, 1994; VanAntwerp and Braatz, 2000; Vandenberghe and Balakrishnan, 1997) or linear matrix inequalities (LMIs) (Boyd *et al.*, 1994; Packard *et al.*, 1991). These two problem formulations differ from each other. BMIs introduce a general framework to formulate control problems, but they are known to be $\mathcal{NP}$-hard (Toker and Ozbay, 1995), and hence no polynomial-time algorithm exists to solve them. On the other hand, not all problems can be formulated in terms of LMIs, but are solved with great practical and theoretical efficiency using interior-point algorithms (Nesterov and Nemirovskii, 1994) which are polynomial-time algorithms. It turns out that, in practice, that is even more effective. Therefore, LMI methods seem to be especially attractive when dealing with n-D systems and LRPs. However, this approach only results in sufficient conditions for stability and requires the problem to be formulated in terms of LMIs which is a difficult task. Although some results on converting stability conditions of n-D systems and LRPs into an LMI form have been published (Du and Xie, 2002; Gałkowski *et al.*, 2002d; Lu, 2002), it is necessary to provide the results for various classes of n-D systems, especially for these where uncertainties, disturbances and delays may appear.

One of the most challenging problems in control theory of 1-D and n-D systems are those related to analysis and designing systems in the presence of uncertainties. It is a critical issue because physical parameter values are only approximately known or vary in time (Ackerman, 1997; Zhou *et al.*, 1996). Another cause of uncertainty is the imperfect knowledge of some system components, or the alteration of their behaviour due to changes in operating conditions. Finally, we have to consider the finite precision of computational issues, which can be, for example introduced by a finite wordlength of the Analog to Digital (A/D) converters and the Digital to Analog (D/A) converters. For these reasons, there is a need to find a method that makes it possible to determine whether or not a system is robustly stable. Unfortunately, the problem of providing such a method that can deal with uncertainties of all sorts, is still unsolved. Moreover, most robust problems are proven to be $\mathcal{NP}$-hard (Blondel and Tsitsiklis, 1997, 2000b). In addition to this, the most common situation encountered when dealing with robust problems is the fact that the controller parameterization depends explicitly on the state-space matrices of the system, which are unknown. This immediately leads to a problem formulation in terms of BMIs, which has no effective solution. However, there are several ways to approximate solutions to robust problems using efficient polynomial time algorithms. The most frequently used approach is based on computing upper bounds on parameter values for which the system remains stable. Further, this can be formulated with convex optimization over LMIs (Boyd *et al.*, 1994; El Ghaoui and Niculescu, 1999; Packard *et al.*, 1991). In what follows, this approach can be used even in the area of n-D systems (Du and Xie, 1999b; Gałkowski *et al.*, 2003b) and their classes i.e. repetitive processes (Gałkowski *et al.*, 2003a, 2002a,b). Nevertheless, the works related to the area LRPs or n-D systems provide only some preliminary results which are suitable for a specific class of systems or processes.

Another set of problems, even for 1-D systems, are those related to integration of stability and performance design objectives, but only for some specialized definitions of performance and stability measures. In general, performance optimality does not guarantee robust stability. Hence, performance should be optimized with a robust stability constraint. To this end, optimization of the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms representing a performance measure has been proposed recently. Since most $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems are shown to be $\mathcal{NP}$-hard and have no analytical solution, we are therefore interested in finding a suboptimal solution which is possible in practice. This strategy involves some numerical search techniques such as convex optimization, which is frequently adopted for solving problems in 1-D system theory (Gahinet and Apkarian, 1994; Scherer and Weiland, 2002; Skelton *et al.*, 1998). Recently, some attempts have been made towards the development of convex optimization methods involving LMIs to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ design for 2-D systems (Du and Xie, 2002; Tuan *et al.*, 2002) but to date, no work has been reported on a solution to this design problem with performance requirements for LRPs. Hence, there is a need to provide these results.

It is well known that the increasing expectations of system dynamic performances is accompanied by a selection of adequate models of a system. In the real world, many systems and process dynamics are affected by delays which cannot be neglected and make the control such the systems complicated (Richard, 2003). Examples of systems with delays in biology, chemistry, economics, mechanics, physics, population dynamics, as well as in engineering sciences are included in (Boukas and Liu, 2003; Dugard and Verriest, 1998; Mahmoud, 2000; Malek-Zavarei and Jamshidi, 1987; Niculescu, 2001) and references therein.

For this reason, in the last few years, great research efforts have been spent on the development of analysis and synthesis techniques which, extending results from the field of delay-free systems, may apply to time-delay systems. However, it turns out that many analysis and design tests and procedures are not suitable for computer implementation due to their high computational complexity. For example, the design of stabilising controllers for time-delay systems may be, in general, $\mathcal{NP}$-hard (Toker and Ozbay, 1996). This has motivated us to seek computationally effective methods for their analysis and synthesis even if they only result in approximate solutions.

The point of interest in 1-D time-delay systems is that they can be modelled using 2-D dimensional theory (Loiseau and Brethé, 1997). Since 1-D time delay systems are indeed infinite-dimensional systems (Hale and Lunel, 1993), then it is highly attractive to use 2-D (n-D) tools, which are clearly finite-dimensional, to analyse them (Agathoklis and Foda, 1989; Kamen, 1980).

On the other hand, most of the existing works on time-delay systems are concerned with control problems of 1-D systems, but relatively little has been reported on 2-D linear systems with delays. The study of such systems with delay has been motivated by the fact that time delays, which correspond to transportation or computation times, are encountered e.g. during the processing of visual images which are intrinsically two-dimensional.

In view of the above facts, there is a need to provide a method suitable for treating many problems of analysis and synthesis of 2-D(n-D) system classes in a unified manner. Additionally, the method to be provided must result in a computer numerical package, which makes not only analysis and synthesis considered systems to be automated processes, but a reasonable computational cost of computations (i.e. polynomial-time) has to be maintained, even for large-scale problems.

In the context of an increasing demand for software that will facilitate the tedious tasks of data input for automatic system design and analysis, numerous software packages have been created. Indeed, since packages like Control System Toolbox for Matlab (The Mathworks Inc., 2002) and Control System Professional Suite integrated with Mathematica (Bakshee, 2003) exist, both classical and modern methods are available for automated analysis and control design. Moreover, recently developed numerical techniques can be applied to a large number of previously unsolved problems (or hard to solve) in system control theory. It should also be pointed out that, even though an analytical solution exists, a numerical search method for the same problem might have a lower computational complexity than the analytical solution.

Unfortunately, these packages are only available for 1-D systems and up now there is no commercial software related to n-D systems. Although there exists a Matlab-based package (Gałkowski *et al.*, 2000) to aid control related analysis/design of LRPs, its use is mainly restricted to 1-D representation and simulation capabilities. However, it provides routines for constructing the discrete approximation of a differential LRP, which is an nontrivial task in the case of LRPs and 2-D systems in general. The main reason for the lack of software tools for n-D systems is that the inherent complex structure of such systems resulted in an absence of satisfactory mathematical and numerical tools for solving related analysis and synthesis problems.

It has been indicated that among numerical search techniques, convex and quasiconvex optimization methods (Boyd and Vandenberghe, 2004) which involve

LMIs (Boyd *et al.*, 1994; Dullerud and Paganini, 2000; El Ghaoui and Niculescu, 1999) are one of the most promising and effective tools for the analysis and synthesis of 1-D and n-D systems. Hence, to apply LMI methods as the algorithmic core for a wide variety of problems which have arisen in n-D control theory, appropriate problem formulations in terms of LMIs are required.

Faced with the above facts, it turns out that convex optimization methods which involve LMIs lead to computer implementable analysis and synthesis tests, and can generally offer an escape from some potential difficulties which have arisen in n-D system theory. However, there are still many analysis and synthesis problems for n-D systems and LRPs, which have no LMI formulations. This is especially valid for n-D system classes for uncertainty, disturbances and delays occurrence cases. For these reasons this work is focused on extending recently developed LMIs methods to solve problems which have arisen in the analysis and synthesis of such n-D system classes.

> *The main objective of this work is to convert non-trivial problems in the analysis and synthesis of LRPs and n-D systems into an LMI framework for solving them efficiently with recently developed software packages. In particular, the problem is to apply LMI methods to n-D system classes subject to parameter uncertainties, disturbances and delays. The resulting LMI conditions are implemented as* Matlab *m-files.*

The following thesis can be formulated:

> *Many analysis and synthesis problems of differential and discrete LRPs and generally n-D systems, including very difficult and practically motivated problems where uncertainties, disturbances and delays may appear, can be solved effectively with LMI methods.*

To confirm this thesis, the following problems have been addressed:

Theoretical aspects:

- provision and development of a computer implementable formulations, which involve LMIs, for the following problems:
  - stability and stabilisation of LRPs subjected to parameter uncertainty,
  - controller synthesis with performance requirements (in the form of $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms) for LRPs,
  - guaranteed cost control of LRPs,
  - stability and stabilisation of state delayed 2-D systems,
  - robust stability and robust stabilisation of state delayed 2-D systems,
- formulating optimization procedures which can be used to attenuate the effects of uncertainty and disturbances.

Implementation aspect:

- Implementation of a MATLAB-based package which numerically solves considered class of problems.

Application aspects:

- application of LMI methods to study stability and convergence properties of iterative algorithms such as iterative learning control procedures,
- applications of LMI methods to analyze parallel computing processes.

The following outlines the structure of this dissertation and shortly highlights its contributions.

Chapter 2: This Chapter reviews the state-space representations of the 2-D (n-D) systems and LRPs that are considered. The models with parameter uncertainty and state delays are also presented. Furthermore, computational problems associated with the analysis and synthesis of n-D systems and their classes (e.g. LRPs) are indicated. In particular, it is shown that existing approaches to the analysis and synthesis of LRPs and 2-D(n-D) systems are limited only to a certain type of problem and they may not be suitable for other classes of problems, especially for those where uncertainties, disturbances and delays may appear. Finally, an example is given to illustrate the applicability of 2-D state-space representation to describe realistic problems arising in computer science.

Chapter 3: The aim here is to present some fundamental facts about linear matrix inequality methods which are the main tools used in this dissertation. Computational aspects of using LMI methods and software to solve them are also considered. This chapter also presents some mathematical tools to manipulate matrix inequalities and their applications, to obtain a LMI form for a given non-LMI formulation.

Chapter 4: This Chapter is devoted to providing a concept of computer-aided robust design for uncertain LRPs. It is shown that LMI methods extend the use of computer software to deal with engineering decision problems with uncertainties. The algorithms to design robust and efficient controllers are also provided. Some computational experiments are presented to illustrate the effectiveness of the proposed algorithms.

Chapter 5: This chapter focuses on providing LMI conditions for analysis and synthesis purposes with performance requirements. This is accompanied with $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control theory. Moreover, the guaranteed cost control problem is considered. The resulting conditions are formulated as optimization problems involving LMIs, which allows us to increase the system performance. It is also shown that most of the problems cannot be solved efficiently with classical methods (if a solution exists) or even that there was not an analytical solution.

Chapter 6: Here 2-D state delayed systems are considered. The stability and stabilisation results are established. To facilitate the computation process, LMIs methods are employed. An optimization technique to improve numerical results is described.

The dissertation concludes with an overview of major results and the contribution of this work. Several directions for future research are identified.

Chapter 2

2-D(n-D) LINEAR SYSTEMS AND LRPS

The linear state-space models for considered classes of 2-D (n-D) systems have been the subject of research for over three decades due to their advantage of providing a simple and intuitive way to analysis and synthesis for n-D systems (Du and Xie, 2002; Gałkowski, 2001a; Kung *et al.*, 1977). Although, 2-D state-space models seem to be similar to 1-D state-space models, some essential differences exist between them. One of the major differences between 1-D and 2-D (n-D) state-space models is that in the 2-D (n-D) case these models deal only with the local state in contrast to the global state which preserves all past information as in 1-D case. Therefore, some principal system concepts like stability or controllability must be formulated for both local and global states.

The state-space representations of 2-D (n-D) systems and their classes are described in detail in this chapter. Based on these representations, some basic properties such as stability, robustness and performance bounds are defined and described. Furthermore, it will be shown that checking these n-D systems properties are not generally amenable to an effective algorithmic solution owing to the fact that they are generally characterized as being $\mathcal{NP}$-hard or undecidable problems. Finally, some examples are used to illustrate applications and properties of presented models.

2.1. State-space models of 2-D linear systems

During the last few decades, considerable attention has been devoted to 2-D systems due to their practical importance (Kaczorek, 1985). In particular, since the linear state-space models were introduced in the 1970's the 2-D systems have been studied intensively.

There are several state-space models for 2-D systems introduced by Roesser (Roesser, 1975), Fornasini and Marchesini (Fornasini and Marchesini, 1978), Attasi (Attasi, 1973) and Kurek (Kurek, 1985) which have been generalized to the n-D models later on. These models have been commonly used to describe 2-D (n-D) systems, and to investigate their several properties.

Here, we will only concentrate on the most common state-space models i.e. Roesser model (RM) and Fornasini-Marchesini model (FMM). Originally, RM had been proposed to analyse and control linear iterative circuits, but it can be used in encoding, decoding and image processing, see (Bose, 1982; Bracewell, 1995; Lu and Antoniou, 1992). One of the key features of this model is that the state vector is partitioned into horizontal and vertical components, say x^h and x^v, respectively. An alternative formalism to RM is FMM, which is extensively used in signal processing (Du *et al.*, 2000) and control (Du and Xie, 2002; Xie *et al.*, 2002). It should be noted that FMM is also used to describe delay-differential models, see for example (Tian and Zhang, 2004; Zhang and Deng, 2001). Both models are easily generalized to the general n-D ($n > 2$) case and also to the continuous or hybrid case (Kaczorek, 1994). Moreover, it turns out that RM and FMM are not fully independent of each other therefore one model can be embedded in the second one.

On the other hand, a 2-D framework has proven to be an effective tool in the study of LRPs (Rogers and Owens, 1992) due to their inherent 2-D system structure. This allows us to use 2-D state-space models i.e. RM and FMM for modelling both the differential and discrete LRPs.

The unquestioned popularity of the state-space methods in 1-D and n-D system theory stems from the fact that they are well understood and efficient and stable numerical linear algebra routines exist, such as the singular value decomposition (SVD) required when manipulating state-space models. In what follows, state-space methods are less sensitive to the size of perturbation in entry data (Higham *et al.*, 2004). This is why the numerical packages prefer state-space representations of linear systems by means of matrices and vectors rather than rational matrix functions or polynomials. Furthermore, much 2-D (n-D) system analysis can be done within Lyapunov's framework, which is most naturally performed in the state space. Additionally, it turns out that the Lyapunov matrix can be found by solving a linear system of equations (LSE), algebraic Riccati equations (ARE) (Doyle *et al.*, 1989) and LMIs (Boyd *et al.*, 1994; Skelton *et al.*, 1998) for which polynomial-time algorithms exist. Thus, the state-space methods have a computational advantage over the transfer function approach and they are used throughout this dissertation.

2.1.1. Roesser state-space model

The Roesser state-space model (RM) (Kung *et al.*, 1977; Roesser, 1975) is defined by the following equations

$$\begin{aligned}\begin{bmatrix} x^h(i+1,j) \\ x^v(i,j+1) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} x^h(i,j) \\ x^v(i,j) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix} u(i,j) \\ y(i,j) &= \begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{bmatrix} \begin{bmatrix} x^h(i,j) \\ x^v(i,j) \end{bmatrix} + \boldsymbol{D}u(i,j)\end{aligned} \tag{2.1}$$

In this model i and j are the positive integer valued horizontal and vertical coefficients, $x^h \in \mathbb{R}^{n_h}$ is the vector of horizontally transmitted information, $x^v \in \mathbb{R}^{n_v}$ is the vector of vertically transmitted information, $u \in \mathbb{R}^l$ is the vector of control inputs and $y \in \mathbb{R}^m$ is the output vector.

This state-space formalism (2.1) has been introduced for a linear iterative circuit, which is considered as a spatial system. An iterative circuit is the combination of individual cells, each of which is identical, in a regular pattern (Malakorn, 2003), and where each cell performs a linear transformation, as it is depicted in Fig. 2.1. This type of circuit is used widely in automata and logical circuit theory. From a practical viewpoint, the iterative circuit may be used in encoding, decoding and image processing (Roesser, 1975). In this case, the boundary conditions are given by

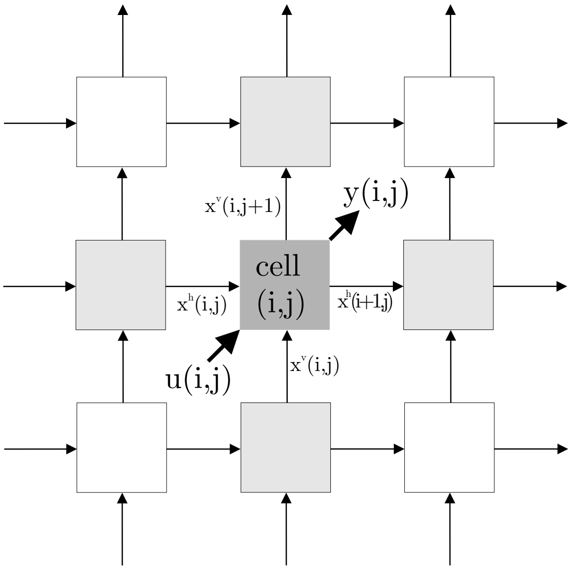


Fig. 2.1. A two-dimensional iterative circuit.

$$\begin{aligned} X_h(0) =& \{x^h(0,j) \quad \forall j : j \geq 0\} \\ X_v(0) =& \{x^v(i,0) \quad \forall i : i \geq 0\} \end{aligned} \tag{2.2}$$

It should be pointed out, that relationships between polynomial matrix theory and state-space description are very strong in the 2-D(n-D) linear case. As a result of application two variable $\mathcal{Z}$ transform in case of RM (2.1), the following transfer function is obtained

$$G^2_{\mathrm{RM}}(z_1, z_2) = \left[\begin{array}{cc} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{array} \right] \left(\left[\begin{array}{cc} \boldsymbol{I} - z_1\boldsymbol{A}_{11} & -z_1\boldsymbol{A}_{12} \\ -z_2\boldsymbol{A}_{21} & \boldsymbol{I} - z_2\boldsymbol{A}_{22} \end{array} \right] \right)^{-1} \left[\begin{array}{c} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{array} \right] + \boldsymbol{D} \tag{2.3}$$

and the characteristic polynomial is given by

$$\mathcal{C}^2_{\mathrm{RM}} = \det \left(\left[\begin{array}{cc} \boldsymbol{I} - z_1\boldsymbol{A}_{11} & -z_1\boldsymbol{A}_{12} \\ -z_2\boldsymbol{A}_{21} & \boldsymbol{I} - z_2\boldsymbol{A}_{22} \end{array} \right] \right) \tag{2.4}$$

2.1.2. Fornasini-Marchesini state-space model

Another commonly used state-space model for 2-D systems is the so-called Fornasini-Marchesini model (FMM) (Fornasini and Marchesini, 1978). The basic model of this type (called the second FMM in literature) has the following form

$$\begin{aligned} x(i+1,j+1) =& \boldsymbol{A}_1 x(i+1,j)+\boldsymbol{A}_2 x(i,j+1)+\boldsymbol{B}_1 u(i+1,j)+\boldsymbol{B}_2 u(i,j+1) \\ y(i,j) =& \boldsymbol{C} x(i,j)+\boldsymbol{D} u(i,j) \end{aligned} \tag{2.5}$$

where, $x(i,j) \in \mathbb{R}^n$ is the local state vector, $u(i,j) \in \mathbb{R}^l$ and $y(i,j) \in \mathbb{R}^m$ are the control input vector and the output vector respectively with $i,j \in \mathbb{N}$. The boundary conditions are defined by

$$\begin{aligned} X_h(0) =& \{x(0,j) \quad \forall j : j \geq 0\} \\ X_v(0) =& \{x(i,0) \quad \forall i : i \geq 0\} \end{aligned} \tag{2.6}$$

This model has been motivated by the algebraic point of view of Nerode equivalence where 2-D input-output map had been factorized (Fornasini and Marchesini, 1976, 1978; Kung *et al.*, 1977). In contrast to RM, the state vector is not split into horizontal and vertical components. Moreover, Fornasini and Marchesini were the first to realize that the major difference between 1-D and 2-D systems is that we can introduce a global and a local state in 2-D (n-D) case. The global state preserves all past information and it is of infinite dimension in general (the diagonal line L_k depicted in Fig. 2.2) while the local state gives us the size of recursion. What is more, the global state, denoted here by X_k, is defined as collection of all local states along $L_k = \{(i,j) :\ i+j = k\}$, hence

$$X_k = \{x(i,j) :\ i+j = k\}$$

In case of FMM of the form (2.5), the transfer function is

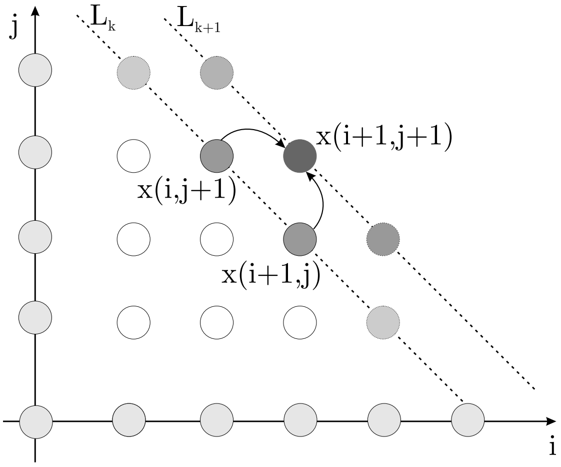


Fig. 2.2. An illustration to FMM and global state.

$$G^2_{\mathrm{FM}}(z_1,z_2) = \boldsymbol{C}\left(\boldsymbol{I} - z_1\boldsymbol{A}_2 - z_2\boldsymbol{A}_1\right)^{-1}\left(z_1\boldsymbol{B}_2 + z_2\boldsymbol{B}_1\right) + \boldsymbol{D} \tag{2.7}$$

and the characteristic polynomial is defined as

$$\mathcal{C}^2_{\mathrm{FM}} = \det\left(\boldsymbol{I} - z_1\boldsymbol{A}_2 - z_2\boldsymbol{A}_1\right) \tag{2.8}$$

2.1.3. State-space models of n-D systems

The models defined in (2.1) and (2.5) can be generalized in an easy way to any $n > 2$. Hence, the n-D system can be described in state space by the Roesser type model of the form

$$\begin{bmatrix} x^1(i_1{+}1,\ldots,i_n) \\ \vdots \\ x^n(i_n,\ldots,i_n{+}1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11} & \ldots & \boldsymbol{A}_{1n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}_{n1} & \ldots & \boldsymbol{A}_{nn} \end{bmatrix} \begin{bmatrix} x^1(i_1,\ldots,i_n) \\ \vdots \\ x^n(i_1,\ldots,i_n) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \vdots \\ \boldsymbol{B}_2 \end{bmatrix} u(i_1,\ldots,i_n)$$
$$y(i_1,\ldots,i_n) = \begin{bmatrix} \boldsymbol{C}_1 & \ldots & \boldsymbol{C}_n \end{bmatrix} \begin{bmatrix} x^1(i_1,\ldots,i_n) \\ \vdots \\ x^n(i_n,\ldots,i_n) \end{bmatrix} + \boldsymbol{D}u(i_1,\ldots,i_n) \tag{2.9}$$

where $x^r(i_1,\ldots,i_n)$, $r = 1,2,\ldots,n$ are the local state sub-vectors, $u(i_1,\ldots,i_n)$ and $y(i_1,\ldots,i_n)$ are the input vector and the output vector respectively with $i_1,\ldots,i_n \in \mathbb{N}$. The characteristic polynomial is associated with (2.9) is then

$$\mathcal{C}^n_{RM}(z_1,\ldots,z_n) = \det\left(\begin{bmatrix} \boldsymbol{I} - z_1\boldsymbol{A}_{11} & \ldots & z_1\boldsymbol{A}_{1n} \\ \vdots & \ddots & \vdots \\ z_n\boldsymbol{A}_{n1} & \ldots & \boldsymbol{I} - z_n\boldsymbol{A}_{nn} \end{bmatrix}\right) \tag{2.10}$$

On the other hand, the Fornasini-Marchesini state-space model for n-D systems is

$$\begin{aligned} x(i_1{+}1,i_2{+}1,\ldots,i_n{+}1) =& \boldsymbol{A}_1 x(i_1,i_2{+}1,\ldots,i_n{+}1) \\ &+\boldsymbol{A}_2 x(i_1{+}1,i_2,i_3{+}1,\ldots,i_n{+}1) \\ &+\ldots+\boldsymbol{A}_n x(i_1+1,i_2+1,\ldots,i_{n-1}+1,i_n) \\ &+\boldsymbol{B}_1 u(i_1,i_2+1,\ldots,i_n+1) \\ &+\ldots+\boldsymbol{B}_n u(i_1+1,i_2+1,\ldots,i_{n-1}+1,i_n) \\ y(i_1,i_2,\ldots,i_n) =& \boldsymbol{C}x(i_1,i_2,\ldots,i_n) + \boldsymbol{D}u(i_1,i_2,\ldots,i_{n-1},i_n) \end{aligned} \tag{2.11}$$

where $x(i_1,i_2,\ldots,i_n)$ is the local state vector, $u(i_1,i_2,\ldots,i_n)$ is the input vector and $y(i_1,i_2,\ldots,i_n)$ is the output vector with $i_1,\ldots,i_n \in \mathbb{N}$. The characteristic polynomial is given by

$$\mathcal{C}^n_{FM}(z_1,\ldots,z_n) = \det\left(\boldsymbol{I} - \sum_{i=1}^{n} z_i\boldsymbol{A}_i\right) \tag{2.12}$$

2.1.4. Relation between models

It is worth mentioning that RM and FMM are not fully independent of each other. To see this, consider the FMM equations (2.5) and define a new state $\xi(i,j)$ as

$$\xi(i,j) = x(i,j+1) - \boldsymbol{A}_1 x(i,j) \tag{2.13}$$

then

$$\begin{aligned}\xi(i+1,j) =&x(i+1,j+1) - \boldsymbol{A}_1(i+1,j) + \boldsymbol{B}u(i,j)\\ =&\boldsymbol{A}_0x(i,j) + \boldsymbol{A}_2\left[\xi(i,j) + \boldsymbol{A}_1x(i,j)\right] + \boldsymbol{B}u(i,j)\\ =&\boldsymbol{A}_2\xi(i,j) + \left[\boldsymbol{A}_0 + \boldsymbol{A}_2\boldsymbol{A}_1\right]x(i,j) + \boldsymbol{B}u(i,j)\end{aligned}$$

Hence

$$\begin{aligned}\begin{bmatrix}\xi(i+1,j)\\ x(i,j+1)\end{bmatrix} &= \begin{bmatrix}\boldsymbol{A}_2 & \boldsymbol{A}_0 + \boldsymbol{A}_2\boldsymbol{A}_1\\ \boldsymbol{I} & \boldsymbol{A}_1\end{bmatrix}\begin{bmatrix}\xi(i,j)\\ x(i,j)\end{bmatrix} + \begin{bmatrix}\boldsymbol{B}\\ \boldsymbol{0}\end{bmatrix}u(i,j)\\ y(i,j) &= \begin{bmatrix}\boldsymbol{0} & \boldsymbol{C}\end{bmatrix}\begin{bmatrix}\xi(i,j)\\ x(i,j)\end{bmatrix} + \boldsymbol{D}u(i,j)\end{aligned}$$

which is identical to the RM form described by (2.1). On the other hand, by identifying in (2.1) the vector

$$x(i,j) = \begin{bmatrix}x^h(i,j)\\ x^v(i,j)\end{bmatrix}$$

and the matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\overline{\boldsymbol{B}}_1$ and $\overline{\boldsymbol{B}}_2$ with

$$\boldsymbol{A}_1 = \begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix},\ \boldsymbol{A}_2 = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22}\end{bmatrix},\ \overline{\boldsymbol{B}}_1 = \begin{bmatrix}\boldsymbol{B}_1\\ \boldsymbol{0}\end{bmatrix},\ \overline{\boldsymbol{B}}_2 = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{B}_2\end{bmatrix}$$

and use simple operations as in (Fornasini and Marchesini, 1978) to obtain the FMM form (2.5).

Throughout this dissertation, the results are presented for one model (RM or FMM) only with the understanding that they could be applied to another model after the above mentioned transformations.

2.2. Linear repetitive processes

Linear repetitive processes (LRPs) are one of the most important classes of 2-D linear systems of both practical and algorithmic interest (Amann *et al.*, 1998; Roberts, 2000a; Rogers *et al.*, 2005; Rogers and Owens, 1992). The essential unique characteristic of such a process is a series of sweeps, termed passes, through a set of dynamics defined over a fixed finite duration known as the pass length. On each pass an output, termed the pass profile, is produced which acts as a forcing function on, and hence contributes to, the dynamics of the next pass profile. This, in turn, leads to the unique control problem for these processes in that the output sequence of pass profiles generated can contain oscillations that increase in amplitude in the pass to pass direction.

To introduce a formal definition, let $\alpha < \infty$ denote the pass length (which is assumed to be constant). Then the pass profile $y_k(p)$, $0 \leq p \leq \alpha$ (p is the independent spatial or temporal variable) generated on the pass k acts a forcing function on, and hence contributes to, the dynamics of the new pass profile $y_{k+1}(p)$, $0 \leq p \leq \alpha$, $k \geq 0$.

The schematic illustration of the dynamics evolution is depicted in Fig. 2.3. This figure also corresponds to the simplest possible case of LRP dynamics where only the previous pass profile contributes to the current one. In this case we deal with unit memory LRPs and within this dissertation, reference will be made only to this subclass of LRPs.

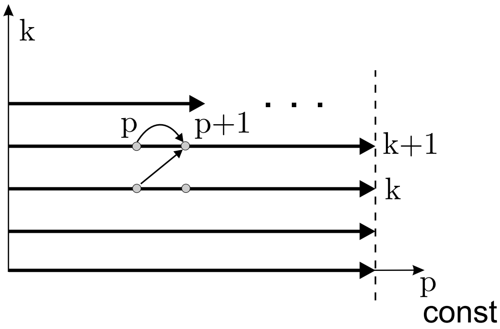


Fig. 2.3 . Schematic illustration of the dynamics of a LRP.

The intrinsic feature of repetitive processes is that their dynamics evolve in two separate directions, i.e.

- from pass to pass direction (k-direction),
- along a given pass of finite duration (p-direction).

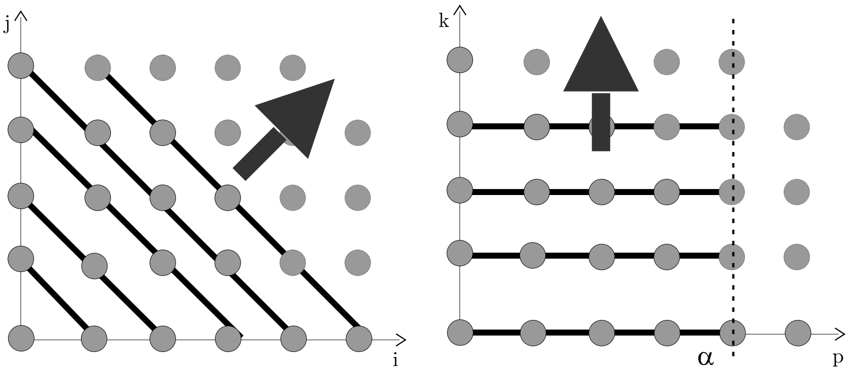
Hence, they clearly have 2-D system structure systems and therefore it is natural to exploit structural links between 2-D linear systems and LRPs. It is worth noting that in the case of LRP, information propagation in one of two separate directions only occurs over a finite duration. This fact is the key difference with other classes of 2-D linear systems (see Fig. 2.4 for illustration). According to the fact that LRP dynamics can evolve as discrete or continuous function of the independent variable (which has temporal or spatial characteristic), two subclasses of LRPs can be considered

- discrete LRPs, where evolution of the dynamics in both directions is discrete,
- differential LRPs where, in contrast to discrete LRPs, the dynamics along the pass evolves as a continuous function of the independent variable (dynamics from pass to pass is still discrete).

The state-space model of a differential LRP has the following, commonly known (Rogers and Owens, 1992), form over $0 \leq t \leq \alpha$, $k \geq 0$

$$\begin{aligned} \dot{x}_{k+1}(t) &= \boldsymbol{A}x_{k+1}(t) + \boldsymbol{B}_0 y_k(t) + \boldsymbol{B}u_{k+1}(t) \\ y_{k+1}(t) &= \boldsymbol{C}x_{k+1}(t) + \boldsymbol{D}_0 y_k(t) + \boldsymbol{D}u_{k+1}(t) \end{aligned} \tag{2.14}$$

Here on pass k, $x_k(t) \in \mathbb{R}^n$ is the state vector, $y_k(t) \in \mathbb{R}^m$ is the pass profile vector, $u_k(t) \in \mathbb{R}^l$ is the vector of control inputs.



A 2-D system An LRP

Fig. 2.4 . Information propagation for 2-D systems and LRPs.

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 $x_{k+1}(0)$ and the initial pass profile (i.e. on the pass number 0) $y_0(t)$. The simplest possible form of them are

$$\begin{aligned} x_{k+1}(0) &= d_{k+1}, \quad k \geq 0, \\ y_0(t) &= f(t), \quad 0 \leq t \leq \alpha \end{aligned} \tag{2.15}$$

where $f(t) \in \mathbb{R}^m$ is a vector whose entries are known functions of t over $[0, \alpha]$ and $d_{k+1} \in \mathbb{R}^n$ is a vector with constant entries.

In case of discrete LRP, the state-space model has the following form over $0 \leq p \leq \alpha$, $k \geq 0$

$$\begin{aligned} x_{k+1}(p+1) =& \boldsymbol{A} x_{k+1}(p) + \boldsymbol{B}_0 y_k(p) + \boldsymbol{B} u_{k+1}(p) \\ y_{k+1}(p) =& \boldsymbol{C} x_{k+1}(p) + \boldsymbol{D}_0 y_k(p) + \boldsymbol{D} u_{k+1}(p) \end{aligned} \tag{2.16}$$

Again it is necessary to specify the boundary conditions, which are given by

$$\begin{aligned} x_{k+1}(0) &= d_{k+1}, \ k \geq 0, \\ y_0(p) &= f(p), \ p = 0, 1, \ldots, \alpha - 1 \end{aligned} \tag{2.17}$$

In some cases (e.g. in the cases of mining or optimal control applications, see (Roberts, 2000a,b; Rogers and Owens, 1992)) the following form of (2.17) are used

$$x_{k+1}(0) = d_{k+1}(0) + \sum_{j=0}^{\alpha-1} \boldsymbol{K}_j y_k(j), \ k = 0, 1, \ldots \tag{2.18}$$

where $\boldsymbol{K}_j$ is a constant matrix of appropriate dimension. These boundary conditions are called dynamic boundary conditions (Gałkowski *et al.*, 2001b; Rogers *et al.*, 2002, 2005). Note that in case of differential LRP, j is sample point along the previous pass.

2.2.1. Linear repetitive processes in terms of Roesser model

It was mentioned that LRPs are a class of 2-D systems thus they share certain structural similarities with 2-D linear systems. Therefore, 2-D state-space models i.e. RM (2.1) and FMM (2.5) can be used for modelling both discrete and differential LRPs. To use the RM, a simple 'forward transformation' of the pass profile vector followed by a change of variable in the pass number are employed. In particular, introduce

$$\begin{aligned} r =& k+1, \\ y_k(p) =& v_{k+1}(p) = v_r(p), \end{aligned} \tag{2.19}$$

then the state-space model (2.16) takes state equation structure in the RM

$$\begin{bmatrix} x_r(p+1) \\ v_{r+1}(p) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix} \begin{bmatrix} x_r(p) \\ v_r(p) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{D} \end{bmatrix} u_r(p) \tag{2.20}$$

Hence in terms of RM, the pass profile vector $y_k(p)$ plays the role of the vertical state vector and the pass state vector $x_{k+1}(p)$ plays the role of horizontal state vector. In what follows, the pass profile vector is simultaneously the output vector, denoted here by $z_r(p)$ and hence we can write

$$z_r(p) = v_r(p) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} x_r(p) \\ v_r(p) \end{bmatrix} \tag{2.21}$$

2.2.2. Linear repetitive processes in terms of Fornasini-Marchesini model

On the other hand, the state-space models of LRPs (2.16) and (2.14) can be embedded into the FMM (2.5). To proceed, define the following matrices from the state-space model (2.16)

$$\widehat{\boldsymbol{A}}_1 = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \widehat{\boldsymbol{A}}_2 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix}, \widehat{\boldsymbol{B}}_1 = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix}, \widehat{\boldsymbol{B}}_2 = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{D} \end{bmatrix} \tag{2.22}$$

and the augmented state vector

$$\hat{x}(p+1, k+1) = \begin{bmatrix} x_k(p+1) \\ y_k(p) \end{bmatrix}$$

Furthermore, introduce the input vector

$$\hat{u}(p+1, k) = \hat{u}(p, k+1) = u_k(p)$$

then the equation (2.16) can be rewritten in the following form

$$\hat{x}(p+1, k+1) = \widehat{\boldsymbol{A}}_1 \hat{x}(p, k+1) + \widehat{\boldsymbol{A}}_2 \hat{x}(p+1, k) + \widehat{\boldsymbol{B}}_1 \hat{u}(p, k+1) + \widehat{\boldsymbol{B}}_2 \hat{u}(p+1, k) \tag{2.23}$$

which is clearly the FMM.

2.2.3. 1-D equivalent model

It is shown (Gramacki, 1999b; Rogers *et al.*, 2005; Rogers and Owens, 1992) that some of the properties of LRPs (e.g. asymptotic stability) can be characterized with other forms of discrete LRPs (also for differential LRPs, after discretization process) using a 1-D equivalent state-space model of the underlying dynamics. Basic steps (for details, see (Gałkowski *et al.*, 2002c; Gramacki, 1999b)) in derivation of a such model involve the change of variables (2.19) and definition of the so-called global state, input and pass profile vectors (α denotes the pass length)

$$\boldsymbol{Y}_r = \begin{bmatrix} \nu_r(0) \\ \nu_r(1) \\ \vdots \\ \nu_r(\alpha-2) \\ \nu_r(\alpha-1) \end{bmatrix}, \boldsymbol{X}_r = \begin{bmatrix} x_r(1) \\ x_r(2) \\ \vdots \\ x_r(\alpha-1) \\ x_r(\alpha) \end{bmatrix}, \boldsymbol{U}_r = \begin{bmatrix} u_r(0) \\ u_r(1) \\ \vdots \\ u_r(\alpha-2) \\ u_r(\alpha-1) \end{bmatrix}$$

Then the 1-D equivalent state-space model for processes described by (2.16) and (2.17) is defined by

$$\begin{aligned} \boldsymbol{Y}(r+1) =& \widetilde{\boldsymbol{\Phi}}\boldsymbol{Y}(r) + \boldsymbol{\Delta}\boldsymbol{U}(r) + \boldsymbol{\Theta}x_r(0) \\ \boldsymbol{X}(r) =& \widetilde{\boldsymbol{\Gamma}}\boldsymbol{Y}(r) + \boldsymbol{\Sigma}\boldsymbol{U}(r) + \boldsymbol{\Psi}x_r(0) \end{aligned} \tag{2.24}$$

where the matrices $\widetilde{\boldsymbol{\Phi}}, \boldsymbol{\Delta}, \boldsymbol{\Theta}$, $\widetilde{\boldsymbol{\Gamma}}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Psi}$ are given in (Gałkowski *et al.*, 2002c; Gramacki, 1999b). It is important to note that use of the presented model allows us to apply 1-D linear system stability tests (in a few very special cases), which can be answered in polynomial time. On the other hand, observing that the matrices dimensions are: $\widetilde{\boldsymbol{\Phi}} \in \mathbb{R}^{m\alpha\times m\alpha}$, $\boldsymbol{\Delta} \in \mathbb{R}^{m\alpha\times l\alpha}$, $\boldsymbol{\Theta} \in \mathbb{R}^{m\alpha\times n}$, $\widetilde{\boldsymbol{\Gamma}} \in \mathbb{R}^{n\alpha\times m\alpha}$ $\boldsymbol{\Sigma} \in \mathbb{R}^{n\alpha\times l\alpha}$, $\boldsymbol{\Psi} \in \mathbb{R}^{n\alpha\times n}$ and they clearly depend on pass length. This fact makes stability tests computationally difficult, especially for large pass length ($\alpha > 20$).

A key feature of the model (2.24) is that along the pass dynamics has been 'hidden' within the global state, input and pass profile vectors which define the equivalent 1-D state-space model. Hence, in 1-D linear system terms, the first equation in (2.24) plays the role of the state vector and the second one is the output equation.

The 1-D model of the form (2.24) is easily extended to the dynamic boundary conditions (2.18) by simply inserting (2.18) into (2.24) to yield

$$\begin{aligned} \boldsymbol{Y}(r+1) =& \boldsymbol{\Phi}\boldsymbol{Y}(r) + \boldsymbol{\Delta}\boldsymbol{U}(r) + \boldsymbol{\Theta}d_r(0) \\ \boldsymbol{X}(r) =& \boldsymbol{\Gamma}\boldsymbol{Y}(r) + \boldsymbol{\Sigma}\boldsymbol{U}(r) + \boldsymbol{\Psi}d_r(0) \end{aligned} \tag{2.25}$$

where

$$\boldsymbol{\Phi} = \widetilde{\boldsymbol{\Phi}} + \boldsymbol{\Theta}\boldsymbol{K}, \quad \boldsymbol{\Gamma} = \widetilde{\boldsymbol{\Gamma}} + \boldsymbol{\Psi}\boldsymbol{K}$$

and

$$\boldsymbol{K} = [\boldsymbol{K}_0\ \boldsymbol{K}_1\ \cdots \boldsymbol{K}_{\alpha-1}]$$

In this case the dimensions of $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$ are equal to the dimensions of $\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{\Gamma}}$ respectively, and the matrix $\boldsymbol{K} \in \mathbb{R}^{n\times m\alpha}$.

2.3. State-space models of 2-D state-delayed systems

To this time, most existing works on time-delay systems deal only with 1-D systems (Boukas and Liu, 2003; Dugard and Verriest, 1998; Mahmoud, 2000; Malek-Zavarei and Jamshidi, 1987; Niculescu, 2001). It is well known that many, or even most physical systems have natural n-D characteristics. It is only for convenience and simplicity, and often to avoid computational complexities, that such features have been neglected.

Due to the fact that time delays correspond to transportation time or computation time, encountered for instance during the processing of a visual image which is intrinsically 2-D (Bracewell, 1995), it becomes appropriate to study 2-D time-delay systems. For this purpose, the state-space representation of 2-D systems with delays in the state are introduced.

2.3.1. Roesser model with state delays

The RM of 2-D discrete linear system with single, possibly different delays along two directions (horizontal and vertical) is represented by

$$\begin{aligned}
\begin{bmatrix} x^h(i{+}1,j) \\ x^v(i,j{+}1) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} x^h(i,j) \\ x^v(i,j) \end{bmatrix} + \begin{bmatrix} \boldsymbol{A}_{11d} & \boldsymbol{A}_{12d} \\ \boldsymbol{A}_{21d} & \boldsymbol{A}_{22d} \end{bmatrix} \begin{bmatrix} x^h(i{-}d_1,j) \\ x^v(i,j{-}d_2) \end{bmatrix} \\
y(i,j) &= \begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{bmatrix} \begin{bmatrix} x^h(i,j) \\ x^v(i,j) \end{bmatrix} + \boldsymbol{D}u(i,j)
\end{aligned} \tag{2.26}$$

where $x^h(i,j) \in \mathbb{R}^{n_1}$ is the horizontal state, $x^v(i,j) \in \mathbb{R}^{n_2}$ is the vertical state. The positive integers d_1 and d_2 represent unknown but constant delays along both directions and satisfy

$$\begin{aligned} 0 \le d_1 \le \overline{d}_1 < \infty \\ 0 \le d_2 \le \overline{d}_2 < \infty \end{aligned}$$

where $\overline{d}_1$ and $\overline{d}_2$ are constant. In this case, the boundary conditions are given by

$$\begin{aligned} X_h(d_1) =& \{x^h(i,j) \quad \forall j \ge 0;\; i = -d_1, -d_1+1, \ldots, 0\} \\ X_v(d_2) =& \{x^v(i,j) \quad \forall i \ge 0;\; j = -d_2, -d_2+1, \ldots, 0\} \end{aligned} \tag{2.27}$$

2.3.2. Fornasini-Marchesini model with state delays

On the other hand, the FMM with state delays has the following form

$$\begin{aligned} x(i+1,j+1) =& \boldsymbol{A}_1 x(i+1,j) + \boldsymbol{A}_2 x(i,j+1) + \boldsymbol{A}_{1d} x(i+1,j-d_1) \\ & + \boldsymbol{A}_{2d} x(i-d_2,j+1) + \boldsymbol{B}_1 u(i+1,j) + \boldsymbol{B}_2 u(i,j+1) \\ y(i,j) =& \boldsymbol{C} x(i,j) + \boldsymbol{D} u(i,j) \end{aligned} \tag{2.28}$$

where $x(i,j) \in \mathbb{R}^n$ is the local state vector, $u(i,j) \in \mathbb{R}^l$ is the input vector, $y(i,j) \in \mathbb{R}^m$ is the output vector and d_1, d_2 are constant positive scalars representing delays along the vertical direction and horizontal direction respectively.

The boundary conditions are given by

$$\begin{aligned} X_h(d_2) =& \{x(i,j) \quad \forall j \geq 0;\ i = -d_2, -d_2+1, \dots, 0\} \\ X_v(d_1) =& \{x(i,j) \quad \forall i \geq 0;\ j = -d_1, -d_1+1, \dots, 0\} \end{aligned} \tag{2.29}$$

2.4. Analysis and synthesis problems in n-D system theory

In this section we are presenting how a multidimensional system structure affects the computational complexity of analysis and synthesis problems for LRPs and generally n-D systems. It should be pointed out that since the frequency domain methods are used some computational difficulties appear which make existing analysis and synthesis tests difficult or impossible to perform on a computer. Furthermore, when it is not feasible to compute an exact solution to a problem, we are interested in finding an approximate solution as this is better than no solution at all. Therefore the alternative or sometimes only existing solutions to some theoretical problems are provided for considered system classes, which allow us to apply numerically reliable design algorithms.

To make computation efficient, the following approach is employed. First, the state-space representation of n-D systems and LRPs, that have been described in previous sections, are used. Next, Lyapunov theory is exploited and the solution to stability and related problems for 2-D(n-D) systems are reduced to the existence positive defined matrix (the Lyapunov matrix). Finally, equipped by this, the analysis and synthesis tests are formulated. These tests seem to be computationally attractive when polynomial-time algorithms for solving Lyapunov equations (inequalities) exist. However, this approach only results in sufficient conditions for stability and usually fails, since we deal with uncertain systems or performance objectives. Therefore it is necessary to find another method which gives us a unifying point of view for most analysis and synthesis problems in n-D system theory.

It sounds attractive to apply LMI methods that have been recognized as a computationally effective tool for solving 1-D system control problems, to solve analysis and synthesis problems in n-D system theory formulated with the state-space representations. This, in turn, makes it possible to create a specialized software tool for automatic n-D system design and analysis.

We will start with the stability problem which is a principal requirement for both LRPs and n-D systems. Next, control objectives and specifications in relation to a stability concept will be considered and control problems will be formulated as a combination of these specifications and the closed-loop representation. Here, we will formulate the control problem in the following form: *Given a set of specifications and an open-loop system, find a controller, if it exists, so that the closed-loop fulfills the given specifications.* This form can be further used to derive LMI formulation for a specific control problem and for efficient computations.

2.4.1. Stability problem

Concept of an asymptotic stability requires that the n-D system represented by the state-space model (2.9) or (2.11) without an external input (i.e. $u(i,j) = 0$)

returns to an equilibrium for any values of boundary conditions (2.2) or (2.6).

It is well known that necessary and sufficient conditions for asymptotic stability of n-D systems can be characterized in terms of a characteristic polynomial of the system matrix.

Lemma 1. *(Jury, 1978) The n-D system is asymptotically stable if and only if the characteristic polynomial given by (2.10) or by (2.12) has no zeros inside the closed unit n-disc, that is,* $\mathcal{C}^n_{RM}(z_1, \ldots, z_n) \neq 0$ *(or* $\mathcal{C}^n_{FM}(z_1, \ldots, z_n) \neq 0$*) for all* $(z_1, \ldots, z_n) \in \overline{U}^n$*, where*

$$\overline{U}^n = \{(z_1, \ldots, z_n) : |z_i| \leq 1 \,, i = 1, \ldots, n\}$$

However, it turns out that this stability test is not computationally efficient and generally the solution is not guaranteed. The main problem here is that zeros of 2-D (n-D) system characteristic polynomial (i.e. system poles) are not isolated as in 1-D case and they cannot be a finite set. Hence, the stability test is not computationally feasible because infinitely many points have to be checked.

Example 2.1. *To see this, consider a 2-D system with the following characteristic polynomial*

$$\rho(z_1, z_2) = 2 - z_1 - z_2 \tag{2.30}$$

It is straightforward to see that all points (z_1, z_2) *that satisfy* $\rho(z_1, z_2) = 0$ *are poles of the system (these points can be:* $\ldots, (-1, 3), (0, 2), (1, 1), (2, 0), (3, -1), \ldots$*). That is, there is the infinite number of system poles which are zeros of (2.30) and they form the line (depicted in Fig. 2.5).*

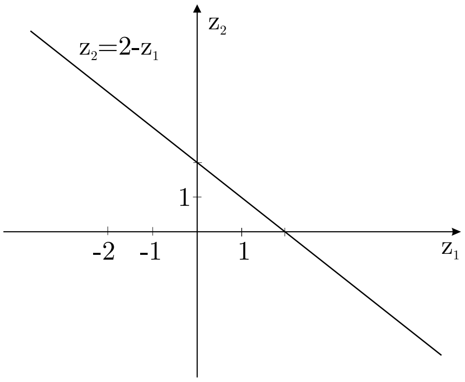


Fig. 2.5 . 2-D system poles.

To make stability tests computationally feasible, it is desirable to provide an approximate solution that results in a significant reduction in computational complexity, but the reduction of computational complexity can only be achieved by introducing some degree of conservativeness i.e. the resulting stability test is only a sufficient one. One of the ways to obtain such stability conditions is by applying Lyapunov theory within state-space models. This approach uses the following candidate of the Lyapunov function

$$V(i_1,\dots,i_n) = x^T(i_1,\dots,i_n)\boldsymbol{P}x(i_1,\dots,i_n)$$

for some $\boldsymbol{P} \succ 0$. Equipped with this, we now have the following Lemma which gives the condition for asymptotic stability of n-D system.

Lemma 2. *(Gałkowski et al., 2003b) Suppose $u(i_1,\dots,i_n) = 0$ for $i_1,...,i_n$ satisfying $i_1+\dots+i_n \geq 0$. If*

$$V_0 = \sum_{i_1+\dots+i_n=0} V(i_1,\dots,i_n) < \infty,$$
$$V_K = \sum_{i_1+\dots+i_n=K} V(i_1,\dots,i_n) < \infty,$$

and

$$V_{K+1} \leq V_K \tag{2.31}$$

then the n-D system represented by the RM (2.9), or by the FMM (2.11) is said to be Lyapunov stable. Moreover, it is asymptotically stable if (2.31) is satisfied for any $K \in \mathbb{N}$ and equality holds in (2.31) only when $x(i_1,\dots,i_n) = 0$.

Above Lemma is a basis for all developments presented in this thesis. It allows us to formulate a sufficient stability condition in terms of LMI.

Lemma 3. *(Gałkowski et al., 2003b) An unforced (i.e. $u(i,j) = 0$) n-D system represented by the RM (2.9) is asymptotically stable if there exist $\boldsymbol{P} \succ 0$, $\boldsymbol{Q}_i \succ 0$, $i = 1,2,\dots,n-1$, satisfying*

$$\begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \\ \vdots \\ \boldsymbol{A}_n^T \end{bmatrix} \boldsymbol{P}\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \cdots & \boldsymbol{A}_n \end{bmatrix} - \begin{bmatrix} \boldsymbol{P}-\sum\limits_{i=1}^{n-1}\boldsymbol{Q}_i & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_1 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_{n-1} \end{bmatrix} \prec 0 \tag{2.32}$$

where the matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, ... ,$\boldsymbol{A}_n$ are identified in (2.9) as

$$\boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{A}_{11} & \cdots & \boldsymbol{A}_{1n} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix}, \dots, \boldsymbol{A}_i = \begin{bmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{A}_{i1} & \cdots & \boldsymbol{A}_{in} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix}, \dots, \boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{A}_{n1} & \cdots & \boldsymbol{A}_{nn} \end{bmatrix}$$

The most important fact associated with Lemma 3 is that an n-D system stability condition can be recast into an LMI feasibility problem i.e. finite dimensional convex optimization problems involving LMI constraints. Indeed, this stability condition requires only to find the finite number of scalar variables. More precisely, since each matrix variable in (2.32) has $\frac{q(q+1)}{2}$ decision variables, then the resulting number of decision variables to be found when solving (2.32) is $n\frac{q(q+1)}{2}$ (q denotes the number of rows (or columns) of the matrix $\boldsymbol{A}_i$).

Although this result can be further used to solve other analysis and synthesis problems of 2-D (n-D) system, to date a smaller number of such problems have been solved in the area of LRPs and n-D systems with delays. In particular, the lack of results is noticeable for differential LRPs. Therefore the main purpose is to provide both effective and applicable stability tests for these processes and systems.

2.4.1.1. Stability of linear repetitive processes

The stability theory of LRPs (Rogers *et al.*, 2005; Rogers and Owens, 1992) is based on an abstract model in a Banach space setting which includes considered LRPs as special cases. This theory consists of two distinct stability concepts i.e.

- asymptotic stability, that guarantees the existence of a limit profile which is described by a 1-D linear system state space model,
- stability along the pass, that guarantees the existence of a limit profile and ensures that the resulting limit profile is stable along the pass dynamics.

In most cases, asymptotic stability is investigated through the use of 1-D system theory applied to the equivalent 1-D model (see Section 2.2.3). However, it turns out that asymptotic stability cannot guarantee that the resulting pass profile has 'acceptable' characteristic and this can be illustrated by the following examples for both differential and discrete cases.

Example 2.2. *Consider the following differential LRP (Benton, 2000), where β is a real scalar,*

$$\begin{aligned}\dot{x}_{k+1}(t) =& - x_{k+1}(t) + u_{k+1}(t) + (1+\beta)\, y_k(t)\\ y_{k+1}(t) =& x_{k+1}(t)\\ x_{k+1}(0) =& 0, \;\; 0 \le t \le \alpha, \;\; k \ge 0.\end{aligned}$$

In this case, the process is asymptotically stable with limit profile over $0 \le t \le \alpha$

$$\begin{aligned}\dot{y}_\infty(t) =& \beta\, y_\infty(t) + u_\infty(t)\\ y_\infty(0) =& 0\end{aligned} \tag{2.33}$$

Also if $u_{k+1}(t) = 1$ and $y_0(t) \equiv 0,\;\; 0 \le t \le \alpha,\;\; k \ge 0$, then it can be easily shown that the first pass profile is given by

$$y_1(t) = 1 - e^{-t}, \;\; 0 \le t \le \alpha. \tag{2.34}$$

But solving the limit profile differential equation (2.33) gives

$$y_\infty(t) = \beta^{-1}(e^{\beta t} - 1),\ \ 0 \leq t \leq \alpha.$$

So although the first pass profile (2.34) is clearly an acceptable dynamic characteristic response to the unit step command $u_1(t) = 1$*, the resulting limit profile has unacceptable dynamic characteristics. In particular, for* $\beta > 0$*, the dynamics of the limit profile increase exponentially and can be said to be 'unstable along the pass' in the obvious intuitive sense.*

Example 2.3. *Consider the discrete LRP of the form (2.16) described by*

$$\begin{aligned} x_{k+1}(p+1) =& -0.5x_{k+1}(p) + (0.5+\beta)y_k(p) + u_{k+1}(p) \\ y_{k+1}(p) =& x_{k+1}(p) \end{aligned}$$

and assume that $x_{k+1}(0) = 0$*. This process is asymptotically stable but the resulting limit profile over* $0 \leq p \leq \alpha$

$$y_\infty(p+1) = \beta y_\infty(p) + u_\infty$$

is unstable in 1*-D sense if* $|\beta| \geq 1$*.*

The reason why asymptotic stability does not guarantee a limit profile which is 'stable along the pass' is the finite pass length. Therefore the strongest concept stability along the pass must be used.

Lemma 4. *(Rogers and Owens, 1992) A differential LRP (2.14) is stable along the pass if and only if the following conditions are satisfied:*

- $\rho(\boldsymbol{D}_0) < 1$
- $\mathrm{Re}(\rho(\boldsymbol{A})) < 0$
- *all eigenvalues of* $G(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}_0$ *with* $s = i\omega$ *have modulus strictly less than unity* $\forall$ *real frequencies* $\omega \geq 0$

where $\boldsymbol{A}$*,* $\boldsymbol{B}_0$*,* $\boldsymbol{C}$*,* $\boldsymbol{D}_0$ *are matrices from the state-space model of LRP (2.14).*

An equivalent set of conditions for stability along the pass can be provided for a discrete LRP too.

Lemma 5. *(Rogers and Owens, 1992) A discrete LRP (2.16) is stable along the pass if and only if the following conditions are satisfied:*

- $\rho(\boldsymbol{D}_0) < 1$
- $\rho(\boldsymbol{A}) < 1$
- *all eigenvalues of* $G(z) = \boldsymbol{C}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}_0$ $\forall |z| = 1$ *have modulus strictly less than unity*

It is clear to see that the third condition is required to make computations for all points on the unit circle. This implies that stability of LRPs problem is an undecidable problem. It is then obvious that there is no possibility of solving such a problem in practice. One of the approaches is to perform computations for finite a set $|z| = 1$ that makes a stability test only a sufficient one. However, it is difficult to determine the number of points for which the computations have to be performed.

Moreover, stability of 1-D equivalent model (2.24) is easily analyzed using the eigenvalues of the matrix $\widetilde{\boldsymbol{\Phi}}$. Unfortunately, this approach cannot be applied in practise when a large number of points on the pass appear because it dramatically increases the computational complexity of the stability tests. Even if it is possible, or there are no such problems, the 1-D model cannot be used in effective robustness and performance analysis because it adds undue complications to the uncertainty and disturbances structure.

To avoid some of these complications, existing structural links between 2-D systems and LRPs (see, Sections 2.2.1 and 2.2.2) will be used to develop stability results for these processes. The fact that information propagation in one of the separate directions occurs over a finite duration, means that existing 2-D system theory can be applied after some modifications.

To proceed, define the shift operators z_1, z_2 in the along the pass (p) and pass-to-pass (k) directions respectively as

$$\begin{aligned} x_k(p) :=& z_1 x_k(p+1), \\ y_k(p) :=& z_2 y_{k+1}(p) \end{aligned}$$

Then the 2-D characteristic polynomial for discrete processes described by (2.16) is defined as

$$\rho(z_1, z_2) = \det\left(\begin{bmatrix} \boldsymbol{I} - z_1\boldsymbol{A} & -z_1\boldsymbol{B}_0 \\ -z_2\boldsymbol{C} & \boldsymbol{I} - z_2\boldsymbol{D}_0 \end{bmatrix}\right)$$

On the other hand, by using the s/z transforms instead of z_1/z_2 one, the characteristic polynomial for processes described by (2.14) is obtained

$$\rho(s, z) := \det\left(\begin{bmatrix} s\boldsymbol{I} - \boldsymbol{A} & -\boldsymbol{B}_0 \\ -z\boldsymbol{C} & \boldsymbol{I} - z\boldsymbol{D}_0 \end{bmatrix}\right)$$

Based on these characteristic polynomials, the stability of both discrete and differential LRP can be investigated.

Lemma 6. *(Gałkowski et al., 2003c) A LRP is stable along the pass if, and only if,*

a)

$$\rho(s, z) \neq 0,\ \forall\ (s, z) : \mathrm{Re}(s) \geq 0, |z| \leq 1$$

in case of differential LRP described by (2.14)

b)

$$\rho(z_1, z_2) \neq 0,\ \forall\ (z_1, z_2) : |z_1| \leq 1, |z_2| \leq 1$$

in case of discrete LRP described by (2.16)

Again, since stability tests are based on computing zeros of 2-D characteristic polynomial then there is no effective method to check them, because in particular, the set of all zeros cannot be finite.

To employ very powerful algorithms for convex optimization based on LMIs, the results developed for 2-D systems are adopted, therefore the following Lyapunov function candidate is used

$$V(k,p) = V_1(k,p) + V_2(k,p) = x_{k+1}^T(p)\boldsymbol{P}_1 x_{k+1}(p) + y_k^T(p)\boldsymbol{P}_2 y_k(p) \tag{2.35}$$

where $\boldsymbol{P}_1 \succ 0$ and $\boldsymbol{P}_2 \succ 0$. It should be pointed out that the above functions are combination of two independent indeterminates due to the 2-D nature of the repetitive processes considered in this dissertation.

In contrary to discrete LRPs, there is no a result for differential case. Indeed there is a need to develop them. It can be done with the following Lyapunov function candidate

$$V(k,t) = V_1(k,t) + V_2(k,t) = x_{k+1}^T(t)\boldsymbol{P}_1 x_{k+1}(t) + y_k^T(t)\boldsymbol{P}_2 y_k(t) \tag{2.36}$$

2.4.1.2. Stability of 2-D state delayed systems

The most natural method to analyse a 2-D system with delays is the transformation of such a system into an equivalent non-delayed system and inspect the augmented matrix. For example, the system represented by (2.28) can be transformed into the following

$$\begin{bmatrix} x(i+1,j+1) \\ x(i-d_2+1,j+1) \\ x(i-d_2,j+1) \\ \vdots \\ x(i-1,j+1) \\ x(i,j+1) \\ x(i+1,i-d_1) \\ x(i+1,j-d_1+1) \\ \vdots \\ x(i+1,j) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_{1d} & \mathbf{0} & \cdots\mathbf{0} & \mathbf{0} & \boldsymbol{A}_{2d} & \mathbf{0}\cdots & \boldsymbol{A}_2 \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \cdots\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots\mathbf{0} & \vdots & \mathbf{0} & \mathbf{0}\cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0}\cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots\mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0}\cdots & \mathbf{0} \\ \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \cdots\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots\mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0}\cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots\mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}\cdots & \mathbf{0} \\ \vdots & \vdots & \ldots\ \vdots & \vdots & \vdots & \vdots & \vdots\ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0}\ \ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\cdots & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} x(i,j+1) \\ x(i-d_2,j+1) \\ x(i-d_2+1,j+1) \\ \vdots \\ x(i-2,j+1) \\ x(i-1,j+1) \\ x(i+1,j-d_1) \\ x(i+1,j-d_1+1) \\ \vdots \\ x(i+1,j) \end{bmatrix}$$

However, immediately some difficulties appear due to the following facts:

- the state delays can be large in both directions, which results in a possibly very large dimension of the augmented matrix,
- if the state delays are not exactly known; the dimension of the augmented matrix is unknown.

Hence, it is very difficult or even impossible in some cases to provide a computationally effective stability test for 2-D state delayed systems in this way. This is the reason why the other methods have to be employed to verify that property, e.g. LMI-based method, which has become a main tool in the analysis and synthesis of 1-D systems (Boukas and Liu, 2003).

Moreover, it is important to note that two types of stability problems for delayed systems exist:

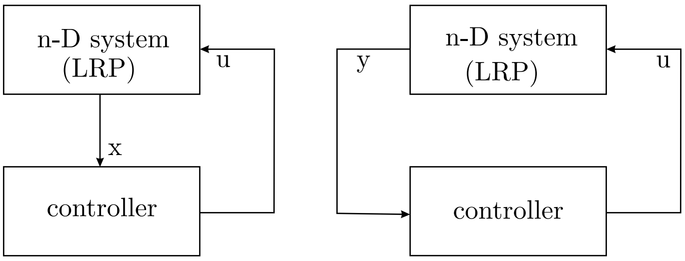
- delay-independent stability (Dugard and Verriest, 1998),
- delay-dependent stability (Lee and Kwon, 2002; Moon *et al.*, 2001).

In the first case, a stability does not depend on delay, which means that systems are stable for any delay but this stability condition is sometimes restrictive. To overcome this drawback, a delay-dependent stability condition has to be provided. It turns out that it is relatively simple to provide a delay-independent stability condition (for some preliminary results for 2-D linear systems with delays, see (Paszke *et al.*, 2003, 2004; Trinh and Fernando, 2000)) but there are problems with formulating delay-dependent stability conditions.

2.4.2. Stabilisation problem

The problem of stabilisation of LRPs and linear 2-D (n-D) systems has received considerable attention in the literature over the last few years (Benton, 2000; Cook, 2000; Du and Xie, 1999a; Lin, 2001; Lin *et al.*, 2001; Rogers and Owens, 1992). The goal of stabilisation is to find a controller connected to the original system so that a stability of the connected system (called closed-loop system - see Fig. 2.6) is guaranteed.

One unique feature of LRPs is that it is possible to define physically meaningful control laws for them. For example, in the ILC application, one such family of control laws is composed of state feedback control action on the current pass combined with information 'feedforward' from the previous pass (or trial in the ILC context) which, of course, has already been generated and is therefore available for use. In the general case of LRPs it is clearly highly desirable to have an analysis setting where such control laws can be designed for stability and guaranteed performance. In this dissertation two types of a controller (connections a plant with a controller) are considered, i.e.



(a) state feedback controller (b) output feedback controller

Fig. 2.6 . Control setups.

(a) static feedback controller (see, Fig. 2.6(a)). In this case, we assume that the state of the system is perfectly available from the state feedback. Hence the following control law can be applied

$$u(i_1,\ldots,i_n)=\begin{bmatrix} \boldsymbol{K}_1 & \cdots & \boldsymbol{K}_n \end{bmatrix}\begin{bmatrix} x^1(i_1,\ldots,i_n) \\ \vdots \\ x^n(i_n,\ldots,i_n) \end{bmatrix} \tag{2.37}$$

(b) dynamic output feedback controller (see, Fig. 2.6(b)). This controller is used when no complete state vector is available then one option is to use an observer to reconstruct it. Therefore the following controller is used

$$\begin{aligned} \begin{bmatrix} x^{c1}(i_1{+}1,\ldots,i_n) \\ \vdots \\ x^{cn}(i_1,\ldots,i_n{+}1) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}_{c_{11}} \ldots \boldsymbol{A}_{c_{1n}} \\ \vdots \ \ \ddots \ \ \vdots \\ \boldsymbol{A}_{c_{n1}} \ldots \boldsymbol{A}_{c_{nn}} \end{bmatrix}\begin{bmatrix} x^{c1}(i_1,\ldots,i_n) \\ \vdots \\ x^{cn}(i_n,\ldots,i_n) \end{bmatrix} \\ &\quad + \begin{bmatrix} \boldsymbol{B}_{c_1} \\ \vdots \\ \boldsymbol{B}_{c_2} \end{bmatrix} y(i_1,\ldots,i_n) \\ u(i_1,\ldots,i_n) &= \begin{bmatrix} \boldsymbol{C}_{c_1} \cdots \boldsymbol{C}_{c_n} \end{bmatrix}\begin{bmatrix} x^{c1}(i_1,\ldots,i_n) \\ \vdots \\ x^{cn}(i_n,\ldots,i_n) \end{bmatrix} \\ &\quad + \boldsymbol{D}_c y(i_1,\ldots,i_n) \end{aligned} \tag{2.38}$$

Note that a static output controller, although possible, has a limited use for n-D systems due to much stronger limitations than for the 1-D case, and the dynamic structure is more frequently investigated.

Now the stabilising control problem can be formulated: *Given the open-loop LRP (or n-D system) find a controller (2.37) or (2.38) (if it exists) such that the closed-loop system is stable along the pass (asymptotically stable).*

The most common situation when we deal with the stabilisation problem is that the condition for controller existence is clearly bilinear in the Lyapunov matrix and the controller matrices, which are variables to be found. Therefore, this problem is generally stated as the BMI problem and hence it probably belongs to the class of $\mathcal{NP}$-hard problems.

2.4.3. Robust control problem

Robustness has recently become one of the most important issues in control systems research (Ackerman, 1997; Dullerud and Paganini, 2000; Zhou *et al.*, 1996). This is the result of taking into consideration uncertainties in the modelling process. When the data is uncertain, there is a need to describe the structure of perturbation which affects system matrices.

In common with 1-D linear systems theory, we consider three models of uncertainty structure which are introduced below. Note that all uncertainties satisfying presented conditions are said to be admissible.

(a) *norm-bounded model* of uncertainty. This model of uncertainty corresponds to a system which matrices uncertainty are modelled as an additive perturbation to the nominal system matrices. Therefore a system is said to be subjected to norm-bounded parameter uncertainty if matrices of such a system can be written in the form

$$\boldsymbol{M} = \boldsymbol{M}_0 + \Delta\boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E} \tag{2.39}$$

where $\boldsymbol{H}$ and $\boldsymbol{E}$ are some known constant matrices with compatible dimensions and $\boldsymbol{M}_0$ defines the nominal system. $\boldsymbol{\mathcal{F}}$ is an unknown, constant matrix which satisfies

$$\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}} \preceq \boldsymbol{I} \tag{2.40}$$

Note, that the above model of uncertainty has been widely adopted in describing parametric uncertainty of 1-D uncertain systems (see (Khargonekar *et al.*, 1990) and the references therein).

(b) *polytopic model* of uncertainty. This model of uncertainty corresponds to a system which matrices range in the polytope of matrices. This means that each system matrix $\boldsymbol{M}$ is only known to lie in a given fix polytope of matrices described by

$$\boldsymbol{M} \in \mathrm{Co}\,(\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_h) \tag{2.41}$$

where Co denotes the convex hull. Then, for positive $i = 1, 2, \ldots, h$, $\boldsymbol{M}$ can be written as

$$\boldsymbol{M} := \left\{ \boldsymbol{X} : \boldsymbol{X} = \sum_{i=1}^{h} \alpha_i \boldsymbol{M}_i, \quad \alpha_i \geq 0, \quad \sum_{i=1}^{h} \alpha_i = 1 \right\}$$

As a simple example, the polytope formed from 4 vertices: $\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3$ and $\boldsymbol{M}_4$ is depicted in Fig. 2.7. It has to be emphasized that polytopic model of uncertainty is only used when we deal with differential equations in the state-space model (2.14). This is motivated by the fact that in the discrete case, the set all stable matrices may not be a convex set.

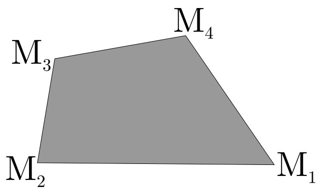


Fig. 2.7. A polytope.

Example 2.4. *Let us consider a polytope formed from* 2 *vertices and assume that they are*

$$\boldsymbol{A}_1 = \begin{bmatrix} 0.5 & 2 \\ 0 & 0.5 \end{bmatrix},\ \boldsymbol{A}_2 = \begin{bmatrix} 0.5 & 0 \\ 2 & 0.5 \end{bmatrix}$$

Based on well known fact that stability in the discrete case is guaranteed if and only if all eigenvalues of a system matrix lie in the interior of the unit circle, it can be seen that the matrices $\boldsymbol{A}_1$ *and* $\boldsymbol{A}_2$ *are stable (*$\lambda_{\max}(\boldsymbol{A}_1) = 0.5$ *and* $\lambda_{\max}(\boldsymbol{A}_2) = 0.5$*). However, a convex combination yields*

$$\boldsymbol{A} = 0.5\boldsymbol{A}_1 + 0.5\boldsymbol{A}_2 = \begin{bmatrix} 0.5 & 1 \\ 1 & 0.5 \end{bmatrix}$$

and $\lambda_{\max}(\boldsymbol{A}) = 1.5$*. This means that* $\boldsymbol{A}$ *is unstable.*

(c) *affine model* of uncertainty. This model of uncertainty corresponds to a system which matrices are modelled as a collection of fixed affine functions of some varying parameters $p_1, \ldots, p_k$ i.e. each matrix can be written in the form

$$\boldsymbol{M}(p) = \boldsymbol{M}_0 + p_1\boldsymbol{M}_1 + \ldots + p_k\boldsymbol{M}_k \tag{2.42}$$

where $\boldsymbol{M}_i\ \forall i = 0, 1, \ldots, k$ are given. Parameter uncertainty is described with range of parameter values. It means that each parameter p_i ranges between two known extremal values $\underline{p_i}$ (minimum) and $\overline{p_i}$ (maximum), therefore it can be written as

$$\underline{p_i} \le p_i \le \overline{p_i}$$

Furthermore, the set of uncertain parameters is

$$\Delta \triangleq \{p{=}(p_1, p_2, \ldots, p_k) : \underline{p_i} \le p_i \le \overline{p_i},\ i{=}1, \ldots, k\}$$

and the set of corners of uncertainty region Δ_0 is defined as

$$\Delta_0 \triangleq \{p{=}(p_1, p_2, \ldots, p_k) : p \in \{\underline{p_i}, \overline{p_i}\},\ i{=}1, \ldots, k\}$$

As an example of a set of uncertain parameters, consider 3 parameters: p_1, p_2, p_3 whose values range in the parameter box formed by their extremal values in 3-D parameter space - see Fig. 2.8.

Imposing one of the uncertainty model on the state-space model of a differential LRP (2.14) (an uncertain discrete model can be written in this form too) the following uncertain state-space model is obtained

$$\begin{aligned} \dot{x}_{k+1}(t) &= \boldsymbol{\mathcal{A}}x_{k+1}(t) + \boldsymbol{\mathcal{B}}_0 y_k(t) + \boldsymbol{\mathcal{B}}u_{k+1}(t) \\ y_{k+1}(t) &= \boldsymbol{\mathcal{C}}x_{k+1}(t) + \boldsymbol{\mathcal{D}}_0 y_k(t) + \boldsymbol{\mathcal{D}}u_{k+1}(t) \end{aligned} \tag{2.43}$$

where the process matrices $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}_0$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathcal{D}}_0$, $\boldsymbol{\mathcal{D}}$ are uncertain and they are described by one of the model defined above.

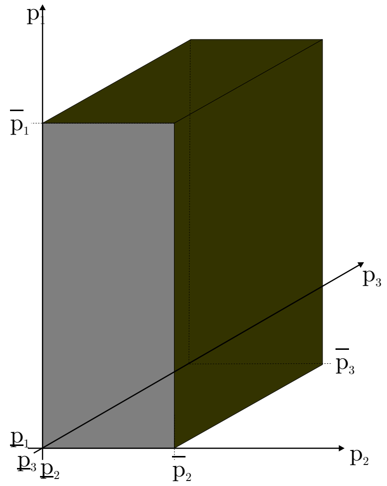


Fig. 2.8 . 3-D parameter space.

Now the robust control problem can be formulated: *Given an uncertain open-loop LRP (or n-D system) find a controller equation (2.37) or (2.38) (if it exists) such that the closed-loop system is stable along the pass (is asymptotically stable) for each admissible system representation.*

It turns out that, in the presence of uncertainty, many classical methods of analysis and synthesis may be limited. This is especially valid for robust control problems in n-D system theory which, similarly to 1-D case, belongs to the class of $\mathcal{NP}$-hard problems because considered problems are nonlinear and nonconvex. This, in turn, is a result of the dependency of the controller matrices on the state-space matrices of the system, which are unknown. Additionally, since any pole assignment method is used, then it is difficult to certify if the poles remain in the prescribed region. Therefore, providing sufficiency conditions, based on approximation techniques, are often about the best we can do. Recently, some preliminary results on applying LMI methods to deal with uncertain systems and processes have been presented (Du and Xie, 1999b; Gałkowski *et al.*, 2003b) but they need to be developed, especially for LRPs.

2.4.4. Control problem with performance requirement

It often turns out that the model which is stable cannot be used in practice. Hence, the ability e.g. to guarantee the upper bounds on signal is more important than the ability to guarantee stability property. For this reason, performance measures and disturbance rejection or attenuation (when stability is ensured) have become important issues in recent years, also for n-D systems and LRPs. It is important to note that disturbance rejection for LRPs and n-D systems have received far less attention than for standard 1-D systems due to a lack of numerically trackable control design methods. This is mainly caused by difficulties in applying pole placement techniques according to the fact that there is no link between the pole placement and dynamic response of the n-D system. Finally, it is difficult to provide an analytical solution for such problems or even know if an analytical solution exists, a numerical search method for the same problem might have a lower computational complexity than the analytical solution.

It is therefore desirable to have computationally effective methods which make it possible to apply popular techniques like the $\mathcal{H}_\infty$ norm (Helton and Merino, 1998; Stoorvogel, 1992; Zhou *et al.*, 1996), the $\mathcal{H}_2$ norm (Saberi *et al.*, 1995), and guaranteed cost control for LRPs and 2-D (n-D) systems (Guan *et al.*, 2001). In particular, the lack of these result is noticeable in the area of differential LRP, there is therefore a need to provide and develop them.

In this dissertation, the effect of disturbances is modelled as an exogenous input w added to the plant model (see Fig. 2.9). Taking this into consideration the additional input results in state-space model (e.g. state-space model of a differential LRP)

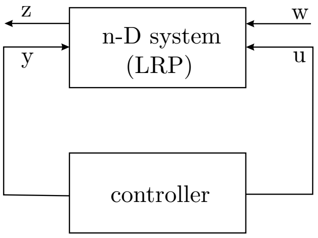


Fig. 2.9 . The setup for performance.

$$\begin{aligned}\dot{x}_{k+1}(t) =& \boldsymbol{A}x_{k+1}(t)+\boldsymbol{B}u_{k+1}(t)+\boldsymbol{B}_0y_k(t)+\boldsymbol{B}_1w_{k+1}(t)\\ y_{k+1}(t) =& \boldsymbol{C}x_{k+1}(t)+\boldsymbol{D}u_{k+1}(t) + \boldsymbol{D}_0y_k(t)+\boldsymbol{D}_1w_{k+1}(t)\end{aligned}$$

Then, in order to quantify the effect of w on output z i.e. pass profile in the case of LRPs, the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms (defined in Chapter 5) are chosen.

Now the control problem with performance requirements can be formulated: *Given an open-loop LRP (n-D system) and performance requirements find a controller (2.37) or (2.38) (if it exists) such that the closed-loop system is stable along the pass (asymptotically stable) and satisfy performance requirements.*

It is clear that this design problem requires performance requirements has to be taken into account, in addition to stability. That is, the main difficulty here is to provide an effective method which can accommodate these objectives into a single design procedure. It turns out that, in the case of 1-D systems, LMI formulation can overcome this difficulty and make the numerical computation process effective. Therefore, due to the lack of results in the area of LRPs, there is a need to find the LMI formulation for the stabilising control problem with performance requirements.

2.5. Applications 2-D systems approach

Many physical processes and systems can be described by embedding them in a 2-D(n-D) framework for further analysis and synthesis (Geng *et al.*, 1990; Kurek and Zaremba, 1993). In addition to this, many, or even most physical systems have natural 2-D (n-D) characteristics but for simplicity, and often to avoid computational complexities, such features have been neglected.

Due to these facts, three engineering examples, from the area of computer sciences, are provided to show use of the 2-D state-space representation and n-D system theory.

2.5.1. Iterative learning control

Iterative learning control (ILC) is a relatively well-known technique for improving the tracking response in systems that repeat a given task or operation over and over again. This technique is motivated by human learning and it is mainly used to learn artificial intelligence systems (e.g. neutral networks) and machines (e.g. robots) and has attracted considerable research interest in recent years (Amann, 1996; Chen and Wen, 1999; Moore, 1983; Owens *et al.*, 2000).

One of the most interesting results of research on ILC is that the learning process can be cast into 2-D framework due to information propagation in two independent directions i.e. time and iterative directions. This, in turn, leads us to apply 2-D stability theory as a useful method for analysis of learning convergence and stability.

To see this, recall that the basic idea of ILC is to use the information from a previous execution of the task in order to improve performance from trial to trial, in the sense that the tracking error is sequentially reduced.

2.5.1.1. Discrete case

To formalize the notion of ILC, let a learning iteration (or a trial), denoted here by the subscript k, be a single execution of the system

$$\begin{aligned} x_k(p+1) =& \boldsymbol{A}x_k(p) + \boldsymbol{B}u_k(p) \\ y_k(p) =& \boldsymbol{C}x_k(p) \end{aligned} \tag{2.44}$$

which has input $u_k(p)$, output $y_k(p)$, and desired output trajectory $y_d(p)$ and where $p \in 1, 2, \ldots, N$. In this case, the boundary conditions are

$$\begin{aligned} x_k(0) =& x_0, \quad k = 0, 1, \ldots \\ u_0(p) =& 0, \quad p = 0, 1, \ldots, N \end{aligned} \tag{2.45}$$

With this notation the following ILC problem is formulated as: find, using a learning technique, an appropriate control sequence $u_k(p)$, $p \in [0, N-1]$ such that tracking error covers to zero along the iterative learning direction (k-direction). In other words, the problem is to derive the optimal input by evaluating the tracking error given by

$$e_k(p) = y_d(p) - y_k(p) \tag{2.46}$$

This is accomplished by adjusting the input from the current trial i.e. $u_k(p)$ to a new input $u_{k+1}(p)$ for the next trial. Therefore, a general iterative control rule can be defined in the following form

$$u_{k+1}(p) = u_k(p) + \Delta u_k(p) \tag{2.47}$$

where $\Delta u_k(p)$ denotes modification of the control input. Now, it follows from (2.44), (2.46) and (2.47) that

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

where

$$\eta_k(p) = x_{k+1}(p-1) - x_k(p-1)$$

Further, from (2.47) and (2.44) can be found

$$\eta_k(p+1) = \boldsymbol{A}\eta_k(p) + \boldsymbol{B}\Delta u_k(p-1)$$

Then, defining the control law

$$\Delta u_k(p) = \boldsymbol{K}_1\eta_k(p+1) + \boldsymbol{K}_2 e_k(p+1)$$

the RM for ILC is obtained

$$\begin{bmatrix} \eta_k(p+1) \\ e_{k+1}(p) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} - \boldsymbol{BK}_1 & -\boldsymbol{BK}_2 \\ -\boldsymbol{CA} + \boldsymbol{CBK}_1 & \boldsymbol{I} - \boldsymbol{CBK}_2 \end{bmatrix} \begin{bmatrix} \eta_k(p) \\ e_k(p) \end{bmatrix}$$

and the boundary condition are given by

$$\begin{aligned} \eta_k(0) =& x_{k+1}(0) - x_k(0) = x_0 - x_0 = 0, \;\; k = 0, 1, \ldots \\ e_0(p) =& y_d(p) - y_0(p) = y_d(p) - \boldsymbol{CA}^T x_0, \;\; p = 0, 1, \ldots, N \end{aligned} \tag{2.48}$$

2.5.1.2. Continuous case

Let us consider the following system

$$\begin{aligned} \dot{x}_k(t) =& \boldsymbol{A}x_k(t) + \boldsymbol{B}u_k(t) \\ y_k(t) =& \boldsymbol{C}x_k(t) \end{aligned} \tag{2.49}$$

where $u_k(t)$ is input, $y_k(t)$ is output, and desired output trajectory is $y_d(t)$. Further, it is assumed that $0 < t \leq T$ where T denotes the known time horizon. The boundary conditions are given by

$$\begin{aligned} x_k(0) =& x_0, \;\; k = 0, 1, \ldots \\ u_0(t) =& f(t), \;\; 0 < t \leq T \end{aligned} \tag{2.50}$$

The aim is to derive the optimal input by evaluating the tracking error

$$e_k(t) = y_d(t) - y_k(t) \tag{2.51}$$

Similarly to the discrete system case, it is accomplished by adjusting the input from the current trial i.e. $u_k(t)$ to a new input $u_{k+1}(t)$ for the next trial. Therefore, a general iterative control rule can be defined in the following form

$$u_{k+1}(t) = u_k(t) + \Delta u_k(t) \tag{2.52}$$

where $\Delta u_k(t)$ denotes modification of the control input. Now, based on (2.49), (2.51) and (2.52) it is seen that

$$e_{k+1}(t) - e_k(t) = -\boldsymbol{CA}\eta_k(t) - \boldsymbol{CB}\int_0^t \Delta u_k(\tau)d\tau$$

where

$$\eta_k(t) = \int_0^t [x_{k+1}(t) - x_k(t)]$$

Further, from (2.52) and (2.49) can be found

$$\frac{d\eta_k(t)}{dt} = \boldsymbol{A}\eta_k(t) + \boldsymbol{B}\int_0^t \Delta u_k(\tau)d\tau$$

Hence, under assumption that $y_d(t)$ is differentiable, the following control law can be defined

$$\Delta u_k(t) = \boldsymbol{K}_1\frac{d\eta_k(t)}{dt} + \boldsymbol{K}_2\frac{de_k(t)}{dt}$$

and the continuous-discrete form of RM for ILC is obtained

$$\begin{bmatrix} \frac{d\eta_k(t)}{dt} \\ e_{k+1}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} - \boldsymbol{BK}_1 & -\boldsymbol{BK}_2 \\ -\boldsymbol{CA} + \boldsymbol{CBK}_1 & \boldsymbol{I} - \boldsymbol{CBK}_2 \end{bmatrix} \begin{bmatrix} \eta_k(t) \\ e_k(t) \end{bmatrix} \tag{2.53}$$

where the boundary conditions are given by

$$\begin{aligned} \eta_k(0) =& 0, \quad k = 0, 1, \dots \\ e_0(t) =& y_d(t) - \boldsymbol{C}e^{\boldsymbol{A}t}x_0 - \int_0^t \boldsymbol{C}e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}u_0(\tau)d\tau \end{aligned}$$

2.5.2. 2-D framework for distributed and parallel computing

Due to a dramatic increase in system complexity and growth of data volume, the problem of efficient computing becomes of crucial importance. One of the options to overcome this problem is to use parallel or distributed computation methods. This, in turn, often requires the use of a networked system. Unfortunately, the effect of network congestion (Srikant, 2004), which naturally arises in data networks such as Internet or ATM and communication delays, can make the result of computations differ from the ideal solution, or make the computation process unstable.

While such situations occur, available data (samples) can be substituted in place of the delayed, or even lost samples, to maintain real time requirements and process stability. Faced with these facts the following problem arises: under what conditions is the stability of the parallel or distributed process guaranteed?

It has been shown in (Bauer *et al.*, 2001) that this problem can be analysed in a 2-D framework. To proceed, it is assumed that the computation process required to evaluate the following difference equation

$$y(n_1, n_2) = \sum_{(i,j)\in\mathcal{M}_0} a_{ij} y(n_1 - i, n_2 - j) + \sum_{(i,j)\in\mathcal{M}_i} b_{ij} x(n_1 - i, n_2 - j) \qquad (2.54)$$

where n_1, n_2 are the positive integer valued coefficients, $a_{ij}, b_{ij} \in \mathbb{R}$, $x(n_1, n_2)$ and $y(n_1, n_2)$ are input and output signals respectively. Further, $\mathcal{M}_i$ and $\mathcal{M}_0$ define input and output masks.

Suppose now that the equation (2.54) is computed in parallel on C processors. Then, in order to ensure computability, inputs within inputs masks and previously computed outputs within outputs masks must be available to each processor in time (processors cannot wait for sample to arrive).

Adapting results and notation from (Bauer *et al.*, 2001) we consider the case when $C = 4$. It is assumed that each processor computes the output for a certain subregion S_i, $i = 1, \ldots, 4$ and they are

$$\begin{aligned} S_1 =& \{(n_1, n_2) : 0 \le n_2 \le 10, n_1 = 0, 4, 8, \ldots\} \\ S_2 =& \{(n_1, n_2) : 0 \le n_2 \le 10, n_1 = 1, 5, 9, \ldots\} \\ S_3 =& \{(n_1, n_2) : 0 \le n_2 \le 10, n_1 = 2, 6, 10, \ldots\} \\ S_4 =& \{(n_1, n_2) : 0 \le n_2 \le 10, n_1 = 3, 7, 11, \ldots\} \end{aligned}$$

In what follows, the processor assignment function f is (f assigns one out of 4 processors to every output sample in the quarterplane)

$$f(n_1, n_2) = \begin{cases} 1 \text{ for } (n_1, n_2) \in S_1 \\ 2 \text{ for } (n_1, n_2) \in S_2 \\ 3 \text{ for } (n_1, n_2) \in S_3 \\ 4 \text{ for } (n_1, n_2) \in S_4 \end{cases}$$

The processors compute the solutions of the following difference equation

$$\begin{aligned} y(n_1, n_2) =& a_{01} y(n_1, n_2 - 1) + a_{10} y(n_1 - 1, n_2) \\ &+ a_{11} y(n_1 - 1, n_2 - 1) + b_{00} x(n_1, n_2) \end{aligned}$$

by mapping the function of the form (this function symbolizes the time instant at which the sample $y(n_1, n_2)$ is computed on the processor $f(n_1, n_2)$)

$$I(n_1, n_2) = \begin{cases} n_2 + 11n_1 \text{ for } (n_1, n_2) \in S_1 \\ n_2 + 2 + 11(n_1 - 1) \text{ for } (n_1, n_2) \in S_2 \\ n_2 + 4 + 11(n_1 - 2) \text{ for } (n_1, n_2) \in S_3 \\ n_2 + 6 + 11(n_1 - 3) \text{ for } (n_1, n_2) \in S_4 \end{cases}$$

As a result we obtain a process for computing the 2-D system response as shown in Fig. 2.10. It is straightforward to see that large delays reduce the amount of

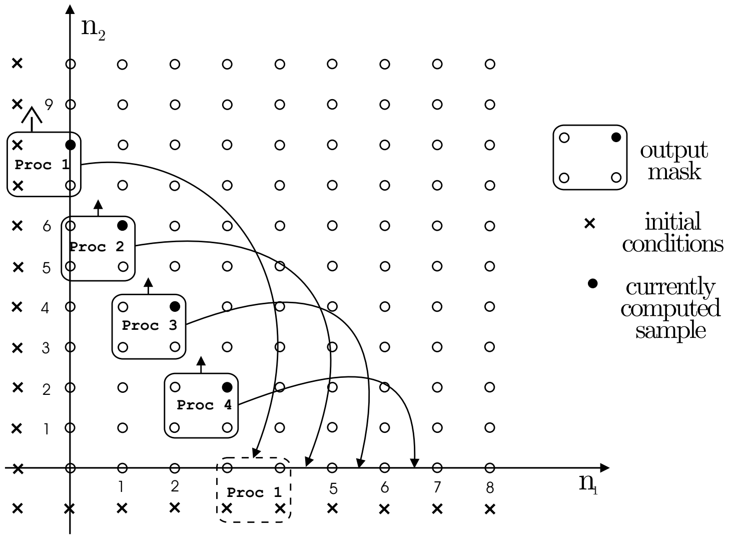


Fig. 2.10 . Order of computation using 4 processors.

parallelism because some samples are not available (see Fig. 2.11). To ensure real time implementation of the equation (2.54), available outputs can be substituted for the missing output samples. Note however, that these substitutions can make

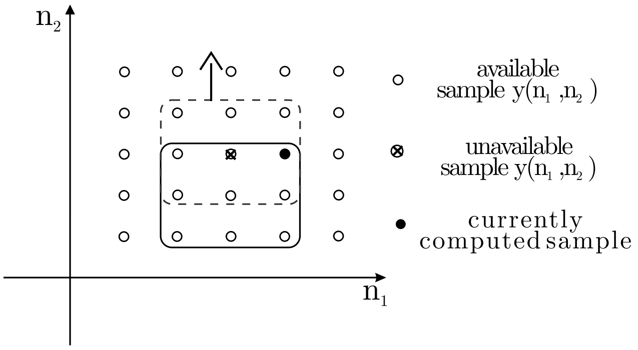

n2
n1
available sample y(n1 ,n2)
unavailable sample y(n1 ,n2)
currently computed sample

Fig. 2.11 . Relationship of available output samples between two consecutive mask positions

our process unstable. To analyse the process behaviour, 2-D system theory can be applied since results on the stability of 2-D systems are available. In order to use such an approach, rewrite (2.54) in terms of RM (2.1) as

$$\begin{bmatrix} x^h(n_1+1,n_2) \\ x^v(n_1,n_2+1) \end{bmatrix} = \boldsymbol{A}(n_1,n_2)x(n_1,n_2) = \begin{bmatrix} \boldsymbol{A}_{11}(n_1,n_2) & \boldsymbol{A}_{12}(n_1,n_2) \\ \boldsymbol{A}_{21}(n_1,n_2) & \boldsymbol{A}_{22}(n_1,n_2) \end{bmatrix} \begin{bmatrix} x^h(n_1,n_2) \\ x^v(n_1,n_2) \end{bmatrix} \tag{2.55}$$

where $x(n_1, n_2) \in \mathbb{R}^{H \times V}$ is the state vector in a 2-D system of order H in n_1 and V in n_2. $\boldsymbol{A}(n_1, n_2)$ is uncertain system matrix which is taken from a set $\boldsymbol{\mathcal{A}}$ which involves all matrices $\boldsymbol{A}(n_1, n_2)$ that correspond to all possible incomplete output masks. Therefore, it seems to be appropriate to make use of developing results on robustness to stability investigation of the system (2.55).

2.5.3. Analysis of iterative algorithms in 2-D system framework

It is well known that most engineering problems are solved on computers with a numerical algorithm. The solution is obtained in an iterative manner where the numerical algorithm is designed to update the solution in each iteration.

One of the important problems solved in this way is the nonlinear optimal control problem based on maximum principle. It is a standard fact that the application of maximum principle, results in a set of nonlinear dynamic equations with mixed boundary conditions.

Roberts (Roberts, 2000a,b) has recognized that the solution search can be cast into 2-D framework where

- one dimension is a time horizon of the dynamic system under investigation,
- and another one is the progress of the iterations.

This allows us to use linear a 2-D state-space model and in turn 2-D system theory to analyse local stability and convergence properties of a specific algorithm designed to achieve the problem solution.

2.5.3.1. Discrete case

Let us consider the real optimal discrete problem (following Roberts (Roberts, 2000b))

$$\min_{u(i)} \left\{ \sum_{i=0}^{N-1} L^*(x(i), u(i), i) \right\}$$

$$\text{subject to} \begin{cases} x(i+1) = f^*(x(i), u(i), i) \\ x(0) = x_0 \end{cases}$$

where $x(i) \in \mathbb{R}^n$ and $u(i) \in \mathbb{R}^l$ are the system state and the control vectors respectively, $L^* : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the real performance function, $f^* : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the real system state equations, and x_0 is a defined initial condition vector. Applications of the maximum principle requires the solution of the following two point boundary value problem

$$\begin{aligned} \widehat{x}(i+1) =& f(\widehat{x}(i), \widehat{p}(i), i), \ \widehat{x}(0) = x_0; \ i \in [0, N-1] \\ \widehat{p}(i+1) =& g(\widehat{x}(i), \widehat{p}(i), i), \ \widehat{p}(N) = 0; \ i \in [0, N-1] \end{aligned}$$

where $p(i) \in \mathbb{R}^n$ is the costate vector and the symbol $\widehat{\ }$ denotes the optimum value. In this case the optimum control is

$$\widehat{u}(i) = h(\widehat{x}(i), \widehat{p}(i), i)$$

The solution of a problem of the above form is obtained in an iterative manner by applying methods such as quasilinearisation or a gradient method in a function space. The general form of such an algorithm can be written in the form

Step 1: Select or compute a nominal solution

$$Y_0^T(i) = \begin{bmatrix} u_0(i) \\ x_0(i) \\ p_0(i) \end{bmatrix}$$

and set the iteration counter $k = 0$

Step 2: Compute an estimate of the optimal control by solving

$$\begin{aligned} X_k(i+1) = \begin{bmatrix} \widehat{x}_k(i+1) \\ \widehat{p}_k(i+1) \end{bmatrix} &= F(X_k(i), Y_k(i)),\ X_k = d_{0_k} \\ \widehat{u}_k(i) =& h(X_k(i)),\ i \in [0, N-1] \end{aligned}$$

Step 3: Update the estimated solution

$$Y_{k+1}(i) = G(X_k(i), Y_k(i))$$

Step 4: Increment $k = k + 1$ and repeat steps 1-3 above until convergence is achieved.

The above algorithm is clearly in the form of a nonlinear discrete repetitive process where d_{0_k} is the initial condition which changes from iteration to iteration and $Y_k(i),\ i \in [0, N]$ acts as a driving input from iteration to iteration. However, analysis of the algorithm can be difficult due to the lack of result on nonlinear repetitive processes. Since we are interested in local convergence properties of the algorithm, appropriate manipulations enable us to write the algorithm in the form of discrete LRPs (2.16)

$$\begin{aligned} X_i(k+1) =& \boldsymbol{A}_0 X_i(k) + \boldsymbol{B}_0 Y_i(k) \\ Y_{i+1}(k) =& \boldsymbol{C} X_i(k) + \boldsymbol{D}_1 Y_i(k) \end{aligned}$$

with

$$X_i(0) = d_i$$

where the pass length $\alpha = N$ and d_i is the vector of initial conditions. In this case, $X_i(k)$ and $Y_i(k)$ are defined as follows

$$X_i(k) = \begin{bmatrix} \widehat{x}_i(k) \\ \widehat{p}_i(k) \end{bmatrix},\ Y_i(k) = \begin{bmatrix} u_i(k) \\ x_i(k) \\ p_i(k+1) \end{bmatrix}$$

and the matrices $\boldsymbol{A}_0$, $\boldsymbol{B}_0$, $\boldsymbol{C}$ and $\boldsymbol{D}_1$ are

$$\boldsymbol{A}_0 = \begin{bmatrix} \boldsymbol{A} + \boldsymbol{B}\overline{\boldsymbol{R}}^{-1}\boldsymbol{B}^T\boldsymbol{A}^{-T}\overline{\boldsymbol{Q}} & \boldsymbol{B}\overline{\boldsymbol{R}}^{-1}\boldsymbol{B}^T\boldsymbol{A}^{-T} \\ -\boldsymbol{A}^{-1}\overline{\boldsymbol{Q}} & \boldsymbol{A}^{-T} \end{bmatrix},$$

$$\boldsymbol{B}_0 = \begin{bmatrix} \boldsymbol{B}^* - \boldsymbol{B}\overline{\boldsymbol{R}}^{-1}\boldsymbol{R}^* & \boldsymbol{A}^* - \boldsymbol{A} - \boldsymbol{B}\overline{\boldsymbol{R}}^{-1}\boldsymbol{B}^T\boldsymbol{A}^{-T}(\overline{\boldsymbol{Q}} - \boldsymbol{Q}^*) & \boldsymbol{B}\overline{\boldsymbol{R}}^{-1}(\boldsymbol{A}^*\boldsymbol{A}^{-1}\boldsymbol{B} - \boldsymbol{B}^*) \\ \boldsymbol{0} & \boldsymbol{A}^{-T}(\overline{\boldsymbol{Q}} - \boldsymbol{Q}^*) & \boldsymbol{I} - (\boldsymbol{A}^*\boldsymbol{A}^{-1})^T \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} k_u\overline{\boldsymbol{R}}^{-1}\boldsymbol{B}^T\boldsymbol{A}^{-T}\overline{\boldsymbol{Q}} & -k_u\overline{\boldsymbol{R}}^{-1}\boldsymbol{B}^T\boldsymbol{A}^{-T} \\ k_x\boldsymbol{I} & \boldsymbol{0} \\ -k_p\boldsymbol{A}^{-T}\overline{\boldsymbol{Q}} & k_p\boldsymbol{A}^{-T} \end{bmatrix},$$

$$\boldsymbol{D}_1 = \begin{bmatrix} \boldsymbol{I} - k_u\overline{\boldsymbol{R}}^{-1}\boldsymbol{R}^* & k_u\overline{\boldsymbol{R}}^{-1}\boldsymbol{B}^T\boldsymbol{A}^{-T}(\overline{\boldsymbol{Q}} - \boldsymbol{Q}^*) & k_u\overline{\boldsymbol{R}}^{-1}(\boldsymbol{A}^*\boldsymbol{A}^{-1}\boldsymbol{B} - \boldsymbol{B}^*)^T \\ \boldsymbol{0} & (1 - k_x)\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & k_p\boldsymbol{A}^{-T}(\overline{\boldsymbol{Q}} - \boldsymbol{Q}^*) & i - k_p\left(\boldsymbol{A}^*\boldsymbol{A}^{-1}\right)^T \end{bmatrix}$$

The matrices $\overline{\boldsymbol{R}}$, $\overline{\boldsymbol{Q}}$, $\boldsymbol{A}$, $\boldsymbol{B}$ represent the current solution and $\boldsymbol{R}^*$, $\boldsymbol{Q}^*$ represent the optimal solution. The parameters k_u, k_x and k_p are scalar gain parameters.

2.5.3.2. Continuous case

Now consider the problem of determining the solution of the real optimal control (Roberts, 2000a)

$$\begin{gathered} \min_{u(t)} \left\{ \phi^*(x(T)) + \int_0^T L^*(x(t), u(t))dt \right\} \\ \text{subject to} \begin{cases} \dot{x}(t) = f^*(x(t), u(t)) \\ x(0) = x_0 \\ \psi^*(x(T)) = 0 \end{cases} \end{gathered} \tag{2.56}$$

defined over finite time horizon $t \in [0, T]$, where $u(t) \in^n \mathbb{R}$and $x(t) \in \mathbb{R}^n$ are continuous control and state vectors respectively, $\phi^* : \mathbb{R}^n \to \mathbb{R}$ is real terminal measure, $L^* : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is real performance measure function, $f^* : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents real system state equations, and $\psi^* : \mathbb{R}^m \to \mathbb{R}^q$ is the real terminal constraint vector.

By carrying out required manipulations presented in (Roberts, 2000a), it is shown that the solution of (2.56) can be found iteratively by solving

$$\begin{aligned} \frac{d}{dt}X^{(i)}(t) =& \boldsymbol{A}_0X^{(i)}(t) + \boldsymbol{B}_0Y^{(i)}(t) \\ Y^{(i+1)}(t) =& \boldsymbol{C}X^{(i)}(t) + \boldsymbol{D}_1Y^{(i)}(t) \end{aligned} \tag{2.57}$$

where

$$X^{(i)}(t) = \begin{bmatrix} \widehat{x}^{(i)}(t) \\ \widehat{p}^{(i)}(t) \end{bmatrix}, \; Y^{(i)}(t) = \begin{bmatrix} u^{(i)}(t) \\ x^{(i)}(t) \\ p^{(i)}(t) \end{bmatrix}$$

and

$$\boldsymbol{A}_0 = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}\overline{\boldsymbol{R}}^{-1}\boldsymbol{B}^T \\ -\overline{\boldsymbol{Q}} & -\boldsymbol{A}^{-T} \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} \boldsymbol{0} & -k_u\overline{\boldsymbol{R}}^{-1}\boldsymbol{B}^T \\ k_x\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & k_p\boldsymbol{I} \end{bmatrix},$$

$$\boldsymbol{B}_0 = \begin{bmatrix} \boldsymbol{B}^* - \boldsymbol{B}\overline{\boldsymbol{R}}^{-1}\boldsymbol{R}^* & \boldsymbol{A}^* - \boldsymbol{A} & \boldsymbol{B}\overline{\boldsymbol{R}}^{-1}(\boldsymbol{B} - \boldsymbol{B}^*)^T \\ \boldsymbol{0} & \overline{\boldsymbol{Q}} - \boldsymbol{Q}^* & (\boldsymbol{A} - \boldsymbol{A}^*)^T \end{bmatrix},$$

$$\boldsymbol{D}_1 = \begin{bmatrix} \boldsymbol{I} - k_u\overline{\boldsymbol{R}}^{-1}\boldsymbol{R}^* & \boldsymbol{0} & k_u\overline{\boldsymbol{R}}^{-1}(\boldsymbol{B} - \boldsymbol{B}^*)^T \\ \boldsymbol{0} & (1 - k_x)\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & (1 - k_p)\boldsymbol{I} \end{bmatrix}$$

which clearly has the form of differential LRPs (2.14). Similarly to the discrete case, the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\overline{\boldsymbol{Q}}$ and $\overline{\boldsymbol{R}}$ represent the current solution, $\boldsymbol{R}^*$, $\boldsymbol{Q}^*$ stand the optimal solution and k_u, k_x and k_p are scalar gain parameters.

2.6. Software for n-D (LRP) analysis and design

The great advantage of the state-space description is that the analysis and synthesis methods derived from it are easily amenable to computer implementation. In what follows, state-space models require manipulation of vectors and matrices, therefore several software packages for engineers can be used for simulating and for evaluating a system's behaviour through the state-space representation. The most widely used software package which supports such a system description is Matlab (The Mathworks Inc., 2004). This package can be easily adopted for simulating, and also for visualization of LRPs or n-D systems.

To date, some packages for simulating and analysis of 2-D (n-D) systems and LRPs which work under Matlab environment have been presented (D'Andrea, 1999; Gałkowski *et al.*, 2000). Unfortunately, they only provide some simple tools for system simulations, analysis and synthesis. The package Lrp Toolbox (Gałkowski *et al.*, 2000) makes use of 2-D and 1-D models of both differential and discrete LRPs for their analysis. The 1-D model of LRP can be very easily implemented on a computer but becomes quickly impractical when the pass length grows. However, in some simple cases, it is possible to exploit the structure of the matrices. One of the major features of the Lrp Toolbox are routines for constructing the discrete approximation of a differential LRP, which is a nontrivial task in the case of LRPs and 2-D systems in general (Gramacki, 1999a). Nevertheless, this package does not involve any tool to analyse and synthesise uncertain LRPs or performance design purposes.

The Multidimensional Systems (MD) Toolbox (D'Andrea, 1999) involves many useful routines for n-D system analysis and design. They may be used to test the stability of n-D systems, to simulate them and to perform control design for these systems. However, this software does not provide tools to solve n-D system theory problems with uncertain data, and there are no routines to manipulate LRPs and 2-D(n-D) state-delayed systems. Even through LRPs analysis and design are possible with 2-D system routines, this package does not exploit LRPs features (e.g. in LRPs case the pass profile vector is simultaneously the output vector) which leads to other algebraic manipulations and to many simplifications in relation to a clear 2-D approach.

2.7. Concluding remarks

This chapter gives a brief introduction to n-D state-space models and their ability to formulate n-D control theory problems. There is no doubt that n-D systems constitute the important class of systems according to a wide variety of applications arising in both theory and practical applications. Motivated by these applications, the interest in n-D systems is growing and there is a need for numerical software to analyse them. However, commonly known numerical methods for system analysis and design which are based on computing system poles, cannot be directly applied to 2-D (n-D) system analysis because of an infinite number of 2-D (n-D) system poles. Therefore, these methods have very high computational cost and they cannot be directly used as a basis to build a software tool for the automatic analysis and synthesis of n-D systems. In order to overcome computational difficulties we propose to put the alternative problems formulations to some theoretical problems for n-D systems and LRPs which results in computationally feasible tests. The proposed approach is based on combining the state-space representations of a considered class of n-D systems with Lyapunov's framework to derive the problem formulation in terms of LMI which are described in the next chapter.

Chapter 3

LINEAR MATRIX INEQUALITY METHODS

Recently, linear matrix inequality (LMI) methods have become popular among researchers from the control community due to their relative simplicity and effective numerical solution. The basic idea of the LMI methods is to approximate a given problem via an optimization problem with a linear objective and LMI constraints. These problems are recognized to belong to the class of $\mathcal{P}$ problems because they are solved with polynomial-time algorithms.

LMI methods are reviewed and discussed in detail in this chapter. First, we need to describe the background information required to understand what LMIs are. Next, the application of modern polynomial-time interior-points methods to solve optimization problems under LMI constraints are described. In particular, it is shown that convex and quasi-convex optimization problems, which involve LMIs, can be adopted as a core to build software to analyse and synthesise 2-D (or n-D in general) systems and LRPs, where uncertainties, disturbances and delays occur. Moreover, some software packages used to solve convex optimization problems are described.

3.1. Linear matrix inequalities

A linear matrix inequality (LMI) is an expression of the form

$$\boldsymbol{F}(x) = \boldsymbol{F}(x)^T = \boldsymbol{F_0} + \sum_{i=1}^{n} x_i \boldsymbol{F_i} \succeq 0 \tag{3.1}$$

where

- $x = (x_1, \ldots, x_n)$, $x \in \mathbb{R}^n$ - is a vector of n real numbers called the decision variables to be found,
- $\boldsymbol{F_0}, \ldots, \boldsymbol{F_n}$ - a given set of real symmetric matrices of equal dimension,
- the inequality symbol $\succeq$ means that the expression (3.1) is nonnegative defined, that is $z^T\boldsymbol{F}(x)z \geqslant 0$, $\forall z \neq 0$, $z \in \mathbb{R}^n$.

The constraint (3.1) is often used to define a minimization problem, which is commonly referred to as a semidefinite program (SDP) (Vandenberghe and Boyd,

1996) of the form

$$\begin{aligned} &\min \ \ c^T x \\ &\text{subject to} \ \ \boldsymbol{F}(x) \succeq 0 \end{aligned} \tag{3.2}$$

where the vector $c \in \mathbb{R}^n$ is given.

It is important to note that LMIs have several intrinsic and attractive features. Firstly, the LMI (3.1) is convex constraint on x (a convex feasibility set). Secondly, while the constraint is matrix inequality instead of a set of scalar inequalities like in linear programming (LP), a much wider class of feasibility sets can be considered. Indeed, for example, LMIs allow us to describe quadric curves like circles (see, Example 3.1). Thirdly, the convex problems involving LMIs can be solved with powerful interior-point methods (Nesterov and Nemirovskii, 1994). In this case "*solved*" means that we can find the vector of the decision variables x that satisfies the LMI, or determine that no solution exists. Sometimes, such solvability problem is called a feasibility problem or an LMI problem in literature. Moreover, the right hand side of (3.1) is symmetric matrix and it is nonnegative defined if, and only if, all its eigenvalues are nonnegative.

Example 3.1. *To confirm that the feasibility set represented by LMI is the convex set, the following inequality is now considered (adopted from (Meinsma, 1997)).*

$$\underbrace{\begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & x_2 \\ x_1 & x_2 & 1 \end{bmatrix}}_{\boldsymbol{F}(x)} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{F}_0} + x_1 \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{\boldsymbol{F}_1} + x_2 \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{\boldsymbol{F}_2} \succeq 0 \tag{3.3}$$

In this case, we see that the feasible set is the interior of the unit disc $\left(\sqrt{x_1^2 + x_2^2} \leq 1\right)$, *that is depicted in Fig. 3.1. Note that the Schur complement (for details see Section 3.5.1) of the block* $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ *in (3.3) gives the equivalent condition*

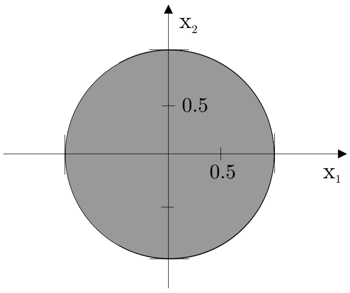


Fig. 3.1 . Example of a feasibility set.

$$1 - \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \succeq 0 \quad \Leftrightarrow \quad 1 - (x_1^2 + x_2^2) \geq 0$$

It is important to note that since the intersection of two or more convex sets is again convex (Boyd and Vandenberghe, 2004), then the intersection of two or more convex sets described by LMIs is again a convex set. This means that there is no distinction between a set of LMIs (where p denotes the number of LMIs)

$$\begin{cases} \boldsymbol{F}^{(1)}(x) \succeq 0 \\ \quad\vdots \\ \boldsymbol{F}^{(p)}(x) \succeq 0 \end{cases}$$

and a single LMI

$$\operatorname{diag}\left(\boldsymbol{F}^{(1)}(x), \ldots, \boldsymbol{F}^{(p)}(x)\right) \succeq 0$$

Example 3.2. *To illustrate the fact that multiple LMIs can be expressed as a single LMI, the following example is considered. Suppose, two convex sets described by (3.3) and*

$$x_1 + 0.5 \geq 0$$

are given. They can be expressed by the LMI of the form

$$\left[\begin{array}{ccc|c} 1 & 0 & x_1 & 0 \\ 0 & 1 & x_2 & 0 \\ x_1 & x_2 & 1 & 0 \\ \hline 0 & 0 & 0 & x_1 + 0.5 \end{array}\right] \succeq 0$$

which represents the convex set depicted in in Fig. 3.2 (the intersection of hyperplane and the interior of the unit circle).

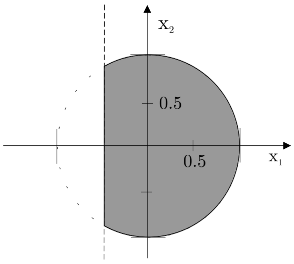


Fig. 3.2 . Example of a feasibility set 2.

It is worth mentioning that in the case of convex optimization problems with convex objective functions, any local minimum is a global minimum. This means that any algorithm that can compute a local minimum for a convex optimization problem, will compute in fact a global minimum.

Remark 3.1. *Note that the expressions* $\boldsymbol{F}(x) \preceq 0$ *and* $\boldsymbol{F}(x) \preceq \boldsymbol{G}(x)$ *can be rewritten as* $-\boldsymbol{F}(x) \succeq 0$ *and* $\boldsymbol{G}(x) - \boldsymbol{F}(x) \succeq 0$ *respectively. Moreover, since an*

LMI is written with $\succeq$ or $\preceq$ then we refer it as a nonstrict LMI. On the other hand, $\prec$ or $\succ$ denotes that the considered LMI is a strict one. It turns out that a nonstrict LMI can be reduced to an equivalent strict LMI by eliminating equality constraints (a more thorough discussion on this can be found in (Boyd et al., 1994)). Keeping these facts in mind, we generally deal with strict negative defined LMIs.

3.1.1. Bilinear matrix inequalities

It is a fact that most control problems cannot be directly written in the form of the LMI (3.1). Thus, bilinear matrix inequalities (BMIs) have been introduced as a general framework to tackle these control problems (Safonov *et al.*, 1994). BMI has the following form

$$\boldsymbol{F}(x,y) = \boldsymbol{F}(x,y)^T = \boldsymbol{F_0} + \sum_{i=1}^{n} x_i \boldsymbol{F_i} + \sum_{j=1}^{m} y_j \boldsymbol{G_j} + \sum_{i=1}^{n}\sum_{j=1}^{m} x_i y_j \boldsymbol{H_{ij}} \succeq 0 \tag{3.4}$$

where the variables are $x = (x_1, \ldots, x_n)$, $x \in \mathbb{R}^n$ and $y = (x_1, \ldots, y_m)$, $y \in \mathbb{R}^m$, and the symmetric matrices $\boldsymbol{F_0}$, $\boldsymbol{F_i}$, $i = 1, \ldots, n$, $\boldsymbol{G_j}$, $j = 1, \ldots, m$ and $\boldsymbol{H_{ij}}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$ are given data. Unfortunately, BMIs are in general highly non-convex optimization problems, which can have multiple local solutions, hence solving a general BMI was shown to be $\mathcal{NP}$-hard (Toker and Ozbay, 1995) i.e. no polynomial-time algorithm has been found so far for solving them (we cannot utilize the convex optimization methods) (Goh *et al.*, 1994). This means that the solution to the problem formulated in terms of BMI can be provided with approximation methods. Obviously, this introduces a significant degree of conservativeness. In effect, BMIs are not so popular as LMIs. However, some of the problems recognized as BMI problems (and therefore assumed to be $\mathcal{NP}$-hard problems) have "hidden" convexity properties and they can be reformulated in terms of LMI.

In this dissertation, a BMI problem formulation will be used to indicate that a considered problem cannot be directly solved with polynomial-time algorithms but in some cases exploration of problem properties can lead to the problem formulation in terms of LMI.

On the other hand, an interesting point to note is that a BMI of the form (3.4) is an LMI in x for fixed y and an LMI in y for fixed x. Furthermore, it means that BMI is convex in x and convex in y but not jointly convex in x and y (VanAntwerp and Braatz, 2000).

Example 3.3. *Consider the following bilinear inequality*

$$1 - xy > 0 \tag{3.5}$$

where x and y are scalar variables. It is clear that (3.5) does not represent a convex set. To see this, we can consider two points on xy-plane which satisfy the constraint (3.5), e.g. $p_1 = (x_1, y_1) = (0.2, 2)$ and $p_2 = (x_2, y_2) = (4, 0.2)$ and we look on the Fig. 3.3. Obviously, the point in the half way between the two values,

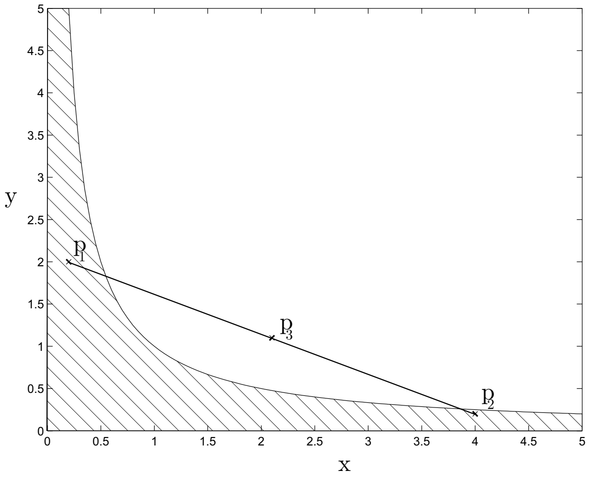


Fig. 3.3 . Example of a non-convex set.

i.e.

$$p_3 = \frac{1}{2}(0.2, 2) + \frac{1}{2}(4, 0.2) = (2.1, 1.1)$$

does not satisfy (3.5).

Today, problems formulated in BMI framework can be solved with algorithms based on a spatial branch and bound strategy which are available only for very small problems. Moreover, there remains some problems in the branch and bound approach therefore these methods are still under development (El Ghaoui and Niculescu, 1999).

3.2. Algorithms and software for LMI methods

Since the simplex algorithm was discovered in the 1940s, much effort has been spent trying to overcome the poor worst-case behaviour of the algorithm - the number of iterations may be exponential in the number of unknowns. This effort resulted in algorithms with better computational complexity. The first polynomial-time algorithm is the ellipsoid algorithm that was developed in the late 1970's by Khachiyan (Khachiyan, 1979). This started an intensive search for algorithms with a strong emphasis on convex optimization and computational complexity issues. Then in 1988, interior point algorithms (or interior point method - IPM) had been established by Nesterov and Nemirovskii. This mainly motivates the increasing popularity of LMI methods (called semi-definite programming - SDP) which are convex optimization problems and can be solved with IPM.

The general features of IPMs are:

- they are efficient in theory - worst-case computational cost of problem solving is polynomial function of the number of decision variables,

- they are efficient in practice i.e. about 5 to 50 iterations, almost independent of input data, are required to find a solution.

Due to the importance of both ellipsoid and interior point algorithms, they will be presented in the next two subsections.

3.2.1. Ellipsoid algorithm

The ellipsoid algorithm is the simplest algorithm to solve LMI problems where the solution with prescribed accuracy is attained in polynomial-time. The basic idea behind this algorithm is to, given a starting ellipsoid that contains the optimal solution (i.e. x^*), cut the ellipsoid in half. Then make a minimal volume ellipsoid enclosing that half that contains the optimal solution, and iterate until the optimum is reached.

In more detail, the ellipsoid algorithm contains the following steps

Step 1: Set the iteration counter $k = 0$ and find a feasible starting point (with standard numerical computations) $x^{(0)}$

Step 2: Compute an ellipsoid $E^{(k)}$ that contains an optimal point $x^{(k)}$

Step 3: Compute a plane that passes through the centre of the ellipsoid so that the solution is guaranteed to lie on one side of the plane

Step 4: Discard that part of the plane which does not contain an optimal point

Step 5: If prescribed accuracy is attained then stop, otherwise set $k = k + 1$ and return to the second step

The above steps guarantee that a sequence of ellipsoids contains an optimal point and the volume of these ellipsoids decreases geometrically (for mathematical details see, for example, (Boyd *et al.*, 1994)). However, the ellipsoid algorithm is rather slow due to slow practical convergence, therefore this algorithm is not particular efficient from a computational point of view. Therefore, available LMI solvers use an interior-point algorithm that currently has many of variations.

3.2.2. Interior-point algorithm

Recently, interior-point methods (IPM) have been applied to optimization problems involving LMIs. The algorithm based on IPM is more computationally efficient in practice than the ellipsoid one (Nesterov and Nemirovskii, 1994; Vandenberghe and Balakrishnan, 1997; Vandenberghe and Boyd, 1996).

The basic idea of the interior-point algorithm is as follows (Meinsma, 1997)

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

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

This algorithm uses the constraints to define a barrier function which is convex within the feasible region and infinite outside it. One of the simplest barrier functions is

$$\phi(x) = -\log\det\left(\boldsymbol{F}(x)\right) = \log\det\left(\boldsymbol{F}^{-1}(x)\right)$$

This barrier function is incorporated into an objective function $f_0(x) = c^T x$ ($c \in \mathbb{R}^n$ is a given vector), which allows the constrained optimization problem to be replaced with an unconstrained optimization problem

$$\min f(x) = \min f_0(x) + \mu\phi(x) = c^T x - \mu \log\det\left(\boldsymbol{F}(x)\right) \tag{3.6}$$

where the parameter $\mu > 0$ is to be selected. The optimization problem (3.6) is iteratively solved with the descent method (Beck, 1991; Henrion, 2003) which produces a minimizing sequence

$$x_{k+1} = x_k + t_k \Delta x_k \tag{3.7}$$

where Δx_k is the step of search direction and $t_k \geq 0$ is the size or step length. It turns out that for the optimization problem (3.6) Newton's method can be applied, therefore t_k is the gradient of $f(x)$, and Δx_k is the inverse of the Hessian of $f(x)$.

As an example of the particular IPM for solving LMIs, we consider the projective method (Gahinet and Nemirovski, 1997; Nemirovskii and Gahinet, 1994), which has been efficiently implemented in a Matlab-based software (Gahinet *et al.*, 1995). This method guarantees to find for $\epsilon > 0$, an ϵ-solution to the problem within a finite number of flops bounded by

$$mn^3 \log\left(\frac{C}{\epsilon}\right)$$

where m is the total row size of the LMIs, n is the total number of decision variables and C is some data-dependent scaling factor. Moreover, no initial feasible solution is required for the projective method.

3.2.3. Implementation and computational issues

It is known that the vector and matrix operations are fundamental to engineering and scientific problems. Therefore, several software packages for engineers provide manipulating and computing on vectors and matrices. Among them, Matlab has become a standard environment for algorithm development and numerical computation, including algorithms for solving problems formulated in terms of LMI. Since most engineering problems are of the form

$$\boldsymbol{F}(x) = \sum_{p=1}^{k} \left[\boldsymbol{L}_p \boldsymbol{X}_p \boldsymbol{R}_p^T + \boldsymbol{R}_p \boldsymbol{X}_p^T \boldsymbol{L}_p^T\right] \tag{3.8}$$

where $\boldsymbol{L}_p, \boldsymbol{R}_p$ are given matrices and $\boldsymbol{X}_p$ is the matrix variable which entries are decision variables $x_1, \ldots, x_n$, then the canonical representation of LMI (3.1) is inconvenient form which makes notation longer. However, each problem written as (3.8) can be formulated in the form (3.1). To see this, the following example is used.

Example 3.4. *Consider the following LMI problem*

$$\boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A} - \boldsymbol{P} \prec 0 \tag{3.9}$$

where $\boldsymbol{P}$ *is the matrix variable of the block diagonal structure (i.e.* $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{P}_h, \boldsymbol{P}_v)$ $\boldsymbol{P}_h \in \mathbb{R}^{n_h \times n_h}$; $\boldsymbol{P}_v \in \mathbb{R}^{n_v \times n_v}$ *and* $n_h + n_v = n$*) to be found and* $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ *is a given matrix (note that the presented LMI is the stability test for a discrete* 2*-D system described by RM (Kaczorek, 1985)).*

Since $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ *(i.e.* $n_h = 1$ *and* $n_v = 1$*) then (3.9) can be expressed in terms of the scalar data and the scalar variables as*

$$\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} - \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix} \prec 0$$

Noting that

$$\boldsymbol{P} = \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix} = x_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = x_1\boldsymbol{P}_1 + x_2\boldsymbol{P}_2$$

we have the following expression

$$x_1 \begin{bmatrix} a_{11}^2 - 1 & a_{11}a_{12} \\ a_{12}a_{11} & a_{12}^2 \end{bmatrix} + x_2 \begin{bmatrix} a_{21}^2 & a_{21}a_{22} \\ a_{22}a_{21} & a_{22}^2 - 1 \end{bmatrix} \prec 0$$

which is in the form of (3.1).

Since the decision variables have a matrix structure, i.e. they are matrices, then it may lead to more efficient computation. To see this fact, observe that the Newton step Δx_k in (3.7) can be found by a solution of the linear system of equations (LSE) written in the form

$$\boldsymbol{H}\Delta x_k = -g \tag{3.10}$$

where

$$\begin{aligned} H_{ij} =& \mathrm{trace}\left(\boldsymbol{F}(x)^{-1}\boldsymbol{F}_i\boldsymbol{F}(x)^{-1}\boldsymbol{F}_j(x)\right) \quad i,j = 1,\ldots,n \\ g_i =& \frac{c_i}{\mu} + \mathrm{trace}\left(\boldsymbol{F}(x)^{-1}\boldsymbol{F}_i\right) \quad i = 1,\ldots,n \end{aligned}$$

The standard approach to solve the LSE defined in (3.10) is to form the Hessian $\boldsymbol{H}$ (which in some cases dominates the cost of solving LSE) and the gradient g, and then use the Cholesky factorization or QR decomposition to find the result. It is important to note that the Cholesky factorization is the most widely used method for solving LSE, but the highest accuracy is obtained only for well-conditioned problems. This means that the method can be used for problems which require less accuracy (the standard accuracy for LMI solvers is 10^{-2}). When the problem becomes ill-conditioned (when we are near to the optimum) the LMI solver switches from the Cholesky factorization to the QR decomposition. In most cases, the QR decomposition is applicable to a wider class of matrices than Cholesky factorization but it requires much greater memory and floating operations. For more details on comparisons between these two methods of solving LSE, see (Golub and Loan, 1996).

Example 3.5. *Consider the problem of computation of the lower bound for stability margin of a* 2*-D system (for further details see Example 3.9). In this example, we aim to emphasize the trade off between accuracy and efficiency (speed) of the solution when a switch from the Cholesky factorization to the QR decomposition occurs. To see this, a random system of the order* $n = 11$ *has been generated and the problem has been solved with the LMI solver. The results are listed in Table 3.1. The LMI solver implemented in* Lmi Control Toolbox *(Gahinet et al., 1995) detects the situation when the Cholesky factorization fails due to numerical instabilities and switches to the QR decomposition. This fact is reported by the message (in this example, it was between* 85th *and* 86th *iteration of the performed simulation).*

Table 3.1. Solutions obtained via the Choleske and the QR decompositions.

accuracy (ϵ)	iterations	QR iterations	CPU time in sec.
10^{-2}	60	0	11.87
10^{-3}	100	15	55.92

```
Solver for generalized eigenvalue minimization
 Iterations       :    Best objective value so far
    .                     .
    85                   0.261484
* switching to QR
    86                   0.261223
    .                     .
```

Note that presented computations have been performed with Lmi Control Toolbox *1.0.8 under* Matlab *6.5. The* Matlab*-files have been run on a PC with AMD Duron 600 MHz CPU and 128MB RAM.*

It should be pointed out that progress in interior point algorithms and hardware technology has been dramatic over the past twenty years. Problems that once took days to solve can now be solved in minutes. The LMI problem with hundreds of variables and constraints, can be solved in a few hours on a reasonably priced desktop workstation or parallel computers, see (Benson, 2003; Lustig and Rothberg, 1996) for some details.

3.2.4. Software for solving LMIs

The growing use of computers and numerical methods result in several software packages for manipulating and for solving LMIs and most of them are available under Matlab (The Mathworks Inc., 2004) and Scilab (Gomez, 1999) environment (but also under the other enviroments, for example the Maple-based package, see (Xhafa and Navarro, 1996)). The most popular packages are: Sp, Sdpa (Fujisawa *et al.*, 2000), Sdpt3 (Toh *et al.*, 2002), Pennon (Ko cvara and Stingl, 2003), Maxdet (Wu *et al.*, 1996), Sdpsol (Wu and Boyd, 1999), Dspd (Benson and Ye, 2002), Mosek (Andersen, 2001), SeDuMi (Sturm, 1999) and Lmi Control Toolbox (Gahinet *et al.*, 1995). Unfortunately, most of them have their own commands to formulate the LMI problem and some of them are designed for specific purposes, e.g. Maxdet is designed for determinant maximization problems, Dsdp exploits structure for combinatorics and Pennon is a package for solving problems of convex and nonconvex nonlinear programming (aimed at large-scale problems with sparse data structure). Moreover, software packages differ from each other on efficiency (see (Mittelmann, 2000) for a recent comparison of these solvers). This is why great attention has been paid to provide uniform interface for specifying LMI. Recently, Yalmip (L ofberg, 2004) has been created which gives us simple commands for rapid prototyping of LMI problems and it supports most of the existing LMI solvers.

Among a large number of available software packages, the following two are considered in this dissertation to deal with LMI problems

- Lmi Control Toolbox,
- SeDuMi with Yalmip for LMI specification.

Both packages use Matlab as the working environment and implement IPM.

3.2.4.1. LMI Control Toolbox

The Lmi Control Toolbox (Gahinet *et al.*, 1995) provides a fully integrated general purpose environment for specifying and solving LMI control problems. This package is mainly destined for 1-D systems but its LMI capabilities make it useful anywhere LMI techniques are applicable, so it can be used for our purposes. Key features of this package are

- support of three LMI solvers: feasibility problems, minimization of linear objectives under LMI constraints, and generalized eigenvalue minimization,
- interactive LMI editor which allows us to specify LMI as symbolic expressions,
- efficient storage and computation of LMI problems.

The Lmi Control Toolbox can handle any system of LMIs of the form (similar to the form represented by 3.8)

$$\boldsymbol{N}^T\boldsymbol{L}(\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_k)\boldsymbol{N} \prec \boldsymbol{M}^T\boldsymbol{R}(\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_k)\boldsymbol{M} \qquad (3.11)$$

where $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k$ are the matrix variables, possibly with some prescribed structure, $\boldsymbol{N}$ and $\boldsymbol{M}$ are known square matrices of identical dimensions and $\boldsymbol{L}(\cdot)$ and $\boldsymbol{R}(\cdot)$ are symmetric block matrices with identical block structures. Each block of $\boldsymbol{L}(\cdot)$ and $\boldsymbol{R}(\cdot)$ is an affine expression in the matrix variables $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k$ and their transposes.

In order to solve an LMI, the following basic steps have to be made

Step 1: To initialize internal representation of the LMIs, the command `setlmis` has to be given as the first step. All data for structure, dimension of the LMIs and matrix variables will be stored in the initialized vector.

Step 2: For each of the unknown matrix variables $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k$ in (3.11), its structure and dimension has to be specified. This is done with the command `lmivar(type,structure)` where `type` selects from available types of matrix variables i.e symmetric, rectangular and other structures. The parameter `structure` gives further information on the structure of $\boldsymbol{X}_i$, $i = 1, 2, \ldots, k$ depending on its type defined by parameter *type*.

Step 3: Specify the term content of each LMI with command `lmiterm`. The syntax is `lmiterm (id, A, B)` where $\boldsymbol{A}$ and $\boldsymbol{B}$ are real matrices, `id=[lmi1, i, j, k]` refers to the $(i,j)th$ block of $\boldsymbol{L}(\cdot)$ or $\boldsymbol{R}(\cdot)$ in (3.11) and `k` denotes the matrix variable number.

Step 4: This is the last step. After declaring the matrix variables and LMIs, the resulting specification is written in the Matlab variable with the command `matlabvariable= getlmis`. Finally, invoke one of the implemented solvers for given problem (one from feasibility, EVP or GEVP problem-see Section 3.3).

Example 3.6. *Consider LMI (3.9) from Example 3.4. Since n_h and n_v are given then $\boldsymbol{A} \in \mathbb{R}^{(n_h+n_v)\times(n_h+n_v)}$ and the solution can be found with the following program written under* Lmi Control Toolbox

```
  setlmis([]);              %Initialize LMI
  P=lmivar(1,[nh 1;nv 1]); % P is the matrix variable
                            % with block-diagonal structure
                            % nh and nv are given

 lmiterm([-1 1 1 P],1,1);   % LMI #1: P>0

 lmiterm([2 1 1 P],A',A);   % LMI #2: A'*P*A
 lmiterm([2 1 1 P],1,-1);   % LMI #2: -P

  example=getlmis;           % LMI name
  feasp(example);            % try to find the solution
                             % by solving feasibility problem
```

3.2.4.2. SeDuMi

SeDuMi (Sturm, 1999) is a Matlab-package which allows us to solve optimization problems with linear, quadratic and semidefiniteness constraints. It implements the primal-dual interior point algorithm (Sturm, 2000). The algorithm has an $\mathrm{O}\left(\sqrt{n}\log\left(\frac{1}{\epsilon}\right)\right)$ worst case bound (n denotes the number of decision variables and ϵ is an accuracy), and treats initialization issues by means of the self-dual embedding technique of (Ye *et al.*, 1994). The iterative solutions are updated in a product form, which makes it possible to provide highly accurate solutions. SeDuMi has some features which are not available under Lmi Control Toolbox

- possibility to impose positive semidefiniteness constraints,
- advantage of matrix sparsity is used to make computation faster,
- the computational worst-case cost of performing each iteration is bounded by $\sqrt{n}\log\left(\frac{1}{\epsilon}\right)$.

Unfortunately, SeDuMi requires the conic formulation of LMI to specify the problem (recall that the conic formulation is widely used by mathematicians) which is inconvenient for our purposes. To work with the standard form of LMI and with matrix variables (see Section 3.2.3 for more details) the pre-processing steps are required (it is clear that these steps are time-consuming). Such conversion can be performed with Yalmip.

3.2.4.3. Yalmip

Yalmip (Löfberg, 2004) is a Matlab toolbox for rapid prototyping of optimization problems. The package initially focused on SDP, but the latest release extends this scope significantly. Yalmip can now be used for convex linear, quadratic, second order cone and SDP, as well as for non-convex SDP, mixed integer, multi-parametric and geometric programming.

The main advantages of Yalmip are:

- only 3 new commands are needed to formulate LMIs,
- constraints and objective functions are defined with an intuitive and standard Matlab code,
- supports numerous (approximately 20) external SDP solvers.

Example 3.7. *In order to solve the LMI problem described in Example 3.4 under* Yalmip *and* SeDuMi*, the following set of commands have to be performed (it is assumed that* $n_h = 1$ *and* $n_v = 1$*)*

```
P=sparse(1:2,1:2,sdpvar(2,1));    % the matrix P
F=lmi(P>0);                       % LMI #1
F=lmi(A'*P*A-P<0);                % LMI #2
solution=solvesdp(F);             % solve optimization problem
```

3.3. Standard LMI problems

The LMI software can solve the LMI problems formulated in three different forms:

- feasibility problem,
- linear optimization problem,
- generalized eigenvalue minimization problem.

3.3.1. Feasibility problem

A feasibility problem is defined as follows

Definition 3.1. *Find a solution $x = (x_1, \ldots, x_n)$ such that*

$$\boldsymbol{F}(x) \succeq 0 \tag{3.12}$$

or determine that the LMI (3.12) is infeasible.

A typical situation for the feasibility problem is a stability problem where one has to decide if a system is stable or not (an LMI is feasible or not). As an example of a feasibility problem, consider Example 3.4.

3.3.2. Linear objective minimization problem

Another standard LMI problem is minimization of a linear objective function under LMI constraint and it is stated as follows

Definition 3.2. *Minimize a linear function $c^T x$ ($x = (x_1, \ldots, x_n)$), where $c \in \mathbb{R}^n$ is a given vector, subject to an LMI constraint (3.12) or determine that the constraint is infeasible. Thus the problem can be written as*

$$\begin{aligned} \min \ & c^T x \\ \text{subject to } \ & \boldsymbol{F}(x) \succeq 0 \end{aligned}$$

This problem can appear in the equivalent form of minimizing the maximum eigenvalue of a matrix that depends affinely on the variable x, subject to an LMI constraint (this is often called EVP)

$$\begin{aligned} \min \ & \lambda \\ \text{subject to } \ & \lambda \boldsymbol{I} - \boldsymbol{F}(x) \succeq 0 \end{aligned}$$

As an example of using this optimization procedure, the problem of the $\mathcal{H}_\infty$ norm computation for a 2-D discrete system is considered.

Example 3.8. *Consider the 2-D system represented by the FMM of the form*

$$\begin{aligned} x(i+1, j+1) =& \boldsymbol{A}_1 x(i+1, j) + \boldsymbol{A}_2 x(i, j+1) + \boldsymbol{B}_1 u(i+1, j) + \boldsymbol{B}_2 u(i, j+1) \\ y(i, j) =& \boldsymbol{C} x(i, j) + \boldsymbol{D} u(i, j) \end{aligned}$$

where the matrices $\boldsymbol{A}_1$*,* $\boldsymbol{A}_2$*,* $\boldsymbol{B}_1$*,* $\boldsymbol{B}_2$*,* $\boldsymbol{C}$ *and* $\boldsymbol{D}$ *are given. Next, introduce the matrices*

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix}, \; \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 \end{bmatrix}, \; \widehat{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C} \end{bmatrix}, \; \widehat{\boldsymbol{D}} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} \end{bmatrix}$$

and

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \; \boldsymbol{R} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{Q} \end{bmatrix}$$

where $\boldsymbol{P} \succ 0$ *and* $\boldsymbol{Q} \succ 0$ *are matrices to be found. Adapting results from (Du and Xie, 2002; Xie et al., 2002), it can be shown that the problem of finding the minimal* $\mathcal{H}_\infty$ *norm bound* γ *is linear objective minimization*

$$\begin{aligned} &\min_{\boldsymbol{P}\succ 0, \boldsymbol{Q}\succ 0} \; \mu \\ &\text{subject to} \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A} + \boldsymbol{R} - \boldsymbol{S} + \boldsymbol{C}^T\boldsymbol{C} & \widehat{\boldsymbol{C}}^T\widehat{\boldsymbol{D}} + \boldsymbol{A}^T\boldsymbol{P}\boldsymbol{B} \\ \boldsymbol{B}^T\boldsymbol{P}\boldsymbol{A} + \widehat{\boldsymbol{D}}^T\widehat{\boldsymbol{C}} & \widehat{\boldsymbol{D}}^T\widehat{\boldsymbol{D}} - \mu\boldsymbol{I} + \boldsymbol{B}^T\boldsymbol{P}\boldsymbol{B} \end{bmatrix} \prec 0 \end{aligned}$$

where $\mu = \gamma^2$*. Since the matrices* $\boldsymbol{A}_1$*,* $\boldsymbol{A}_2$*,* $\boldsymbol{B}_1$*,* $\boldsymbol{B}_2$*,* $\boldsymbol{C}$ *and* $\boldsymbol{D}$ *are assumed to be*

$$\begin{aligned} &\boldsymbol{A}_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; \boldsymbol{A}_2 = \begin{bmatrix} 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; \boldsymbol{B}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \; \boldsymbol{B}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix}, \\ &\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \; \boldsymbol{D} = [0] \end{aligned} \tag{3.13}$$

then the computation performed with LMI CONTROL TOOLBOX *gives us the result* $\gamma = 10.1985$ *(with accuracy* $\epsilon \leq 10^{-2}$*). This result is close to the* $\mathcal{H}_\infty$ *norm of the system* $\gamma = 10.0$ *obtained in an analytic way (Du and Xie, 2002). This means that LMI approach to the* $\mathcal{H}_\infty$ *norm computation of a* 2*-D system allows us to provide not conservative conditions, at least for this example.*

3.3.3. Generalized eigenvalue problem

The last LMI problem is the generalized eigenvalue problem (GEVP) which allows us to minimize the maximum generalized eigenvalue of a pair of matrices that depend affinely on the variable $x = (x_1, \ldots, x_n)$. The general form of GEVP is stated as follows

$$\begin{aligned} &\min \; \lambda \\ &\text{subject to} \; \begin{cases} \boldsymbol{A}(x) \prec \lambda\boldsymbol{B}(x) \\ \boldsymbol{B}(x) \succ 0 \\ \boldsymbol{C}(x) \prec \boldsymbol{D}(x) \end{cases} \end{aligned} \tag{3.14}$$

where $\boldsymbol{C}(x) \prec \boldsymbol{D}(x)$ and $\boldsymbol{A}(x) \prec \lambda\boldsymbol{B}(x)$ denote set of LMIs. It is necessary to distinguish between the standard LMI constraint, i.e.

$$\boldsymbol{C}(x) \prec \boldsymbol{D}(x)$$

and the LMI involving λ (called the linear-fractional LMI constraint)

$$\boldsymbol{A}(x) \prec \lambda \boldsymbol{B}(x)$$

which is quasi-convex with respect to the parameters x and λ. However, this problem can be solved by similar techniques as those for previous problems, see (Boyd *et al.*, 1994) for details.
As an example of using GEVP, let us consider the problem of lower bound for a stability margin computation of a 2-D system.

Example 3.9. *The lower bound for stability margin σ_2 can be obtained by solving the following quasi-convex optimization problem (Xu et al., 2004)*

$$\max_{\boldsymbol{X}\succ 0,\, \boldsymbol{X}_1\succ 0,\, \sigma_2>0} \sigma_2$$
$$\text{subject to} \quad \begin{bmatrix} \widehat{\boldsymbol{A}}_1^T \boldsymbol{X}\widehat{\boldsymbol{A}}_1+\boldsymbol{X}_1-\boldsymbol{X} & (1+\sigma_2)\,\widehat{\boldsymbol{A}}_1^T\boldsymbol{X}\widehat{\boldsymbol{A}}_2 & \boldsymbol{0} \\ (1+\sigma_2)\,\widehat{\boldsymbol{A}}_2^T\boldsymbol{X}\widehat{\boldsymbol{A}}_1 & -\boldsymbol{X}_1 & (1+\sigma_2)\,\widehat{\boldsymbol{A}}_2^T\boldsymbol{X} \\ \boldsymbol{0} & (1+\sigma_2)\,\boldsymbol{X}\widehat{\boldsymbol{A}}_2 & -\boldsymbol{X} \end{bmatrix} \prec 0 \tag{3.15}$$

where the matrices $\widehat{\boldsymbol{A}}_1$, $\widehat{\boldsymbol{A}}_2$ are given and $\boldsymbol{X}\succ 0$, $\boldsymbol{X}_1\succ 0$ are matrix variables to be found. The above optimization problem cannot be directly solved by one of the LMI solver. This is because the condition (3.15) has no form defined by (3.14). Therefore, the following transformations are required. First, decompose the matrix inequality (3.15) on

$$(1+\sigma_2)\begin{bmatrix} \boldsymbol{0} & \widehat{\boldsymbol{A}}_1^T\boldsymbol{X}\widehat{\boldsymbol{A}}_2 & \boldsymbol{0} \\ \widehat{\boldsymbol{A}}_2^T\boldsymbol{X}\widehat{\boldsymbol{A}}_1 & \boldsymbol{0} & \widehat{\boldsymbol{A}}_2^T\boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{X}\widehat{\boldsymbol{A}}_2 & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \widehat{\boldsymbol{A}}_1^T\boldsymbol{X}\widehat{\boldsymbol{A}}_1+\boldsymbol{X}_1-\boldsymbol{X} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{X}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{X} \end{bmatrix} \prec 0$$

which further leads to

$$\begin{bmatrix} \boldsymbol{0} & \widehat{\boldsymbol{A}}_1^T\boldsymbol{X}\widehat{\boldsymbol{A}}_2 & \boldsymbol{0} \\ \widehat{\boldsymbol{A}}_2^T\boldsymbol{X}\widehat{\boldsymbol{A}}_1 & \boldsymbol{0} & \widehat{\boldsymbol{A}}_2^T\boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{X}\widehat{\boldsymbol{A}}_2 & \boldsymbol{0} \end{bmatrix} \prec (1+\sigma_2)^{-1} \begin{bmatrix} -\widehat{\boldsymbol{A}}_1^T\boldsymbol{X}\widehat{\boldsymbol{A}}_1-\boldsymbol{X}_1+\boldsymbol{X} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix}$$

Now, it is obvious that under the following substitutions

$$\lambda = (1+\sigma_2)^{-1}$$

and

$$\boldsymbol{A}(x)=\begin{bmatrix} \boldsymbol{0} & \widehat{\boldsymbol{A}}_1^T\boldsymbol{X}\widehat{\boldsymbol{A}}_2 & \boldsymbol{0} \\ \widehat{\boldsymbol{A}}_2^T\boldsymbol{X}\widehat{\boldsymbol{A}}_1 & \boldsymbol{0} & \widehat{\boldsymbol{A}}_2^T\boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{X}\widehat{\boldsymbol{A}}_2 & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{B}(x)=\begin{bmatrix} -\widehat{\boldsymbol{A}}_1^T\boldsymbol{X}\widehat{\boldsymbol{A}}_1-\boldsymbol{X}_1+\boldsymbol{X} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix}$$

the problem (3.15) can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{X}\succ 0,\ \boldsymbol{X}_1\succ 0,\ \lambda>0} && \lambda \\ &\text{subject to} && \begin{cases} \boldsymbol{A}(x) \prec \lambda \boldsymbol{B}(x) \\ \boldsymbol{B}(x) \succ 0 \end{cases} \end{aligned} \tag{3.16}$$

which is clearly the GEVP of the form (3.14).

Suppose now the 2-D system represented by RM of the form (2.1) is given where

$$\boldsymbol{A}_{11} = -0.5,\quad \boldsymbol{A}_{12} = -0.395,\quad \boldsymbol{A}_{21} = 1,\quad \boldsymbol{A}_{22} = -0.01$$

and the following matrices

$$\widehat{\boldsymbol{A}}_1 = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},\ \widehat{\boldsymbol{A}}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}$$

are defined. As the result of the computations performed with LMI CONTROL TOOLBOX $\sigma_2 = 0.2820$ *has been obtained(with the accuracy* $\epsilon \le 10^{-2}$*). The comparison of the lower bounds for the stability margin derived by presented method here with previous results is shown in Table 3.2. This comparison shows that the lower bounds for the stability margin obtained in this dissertation are larger than those obtained by using the methods in (Agathoklis, 1988) and (Fernando and Trinh, 1999). This means that the lower bounds for the stability margin in this dissertation are less conservative than those reported in (Agathoklis, 1988) and (Fernando and Trinh, 1999). Moreover, it turns out that the analytic result is* $\sigma_2 = \frac{11}{39} = 0.28205128205128$ *therefore it proves again that the LMI approach allows us to provide not conservative conditions.*

Table 3.2. Comparison of the lower bounds for the stability margin σ_2.

Agathoklis's result (Agathoklis, 1988)	$\sigma_2 = 0.127$
Fernando's and Trinh's result (Fernando and Trinh, 1999)	$\sigma_2 = 0.2500$
LMI result	$\sigma_2 = 0.2820$

It should be pointed out that positivity of the term $\boldsymbol{B}(x)$ (see (3.14) and (3.16)) is required for the well-posedness of the problem and the applicability of polynomial-time methods (Gahinet *et al.*, 1995). Due to this, the problems where the following form of $\boldsymbol{B}(x)$ appears

$$\boldsymbol{B}(x) = \begin{bmatrix} \boldsymbol{B}_1(x) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},\quad \boldsymbol{B}_1(x) \succ 0 \tag{3.17}$$

cannot be solved directly with LMI software. To overcome this difficulty, the constraints

$$\boldsymbol{A}(x) \prec \lambda \boldsymbol{B}(x),\quad \boldsymbol{B}(x) \succ 0$$

can be replaced by

$$\boldsymbol{A}(x) \prec \begin{bmatrix} \boldsymbol{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \boldsymbol{Y} \prec \lambda \boldsymbol{B}_1(x), \quad \boldsymbol{B}_1(x) \succ 0$$

where $\boldsymbol{Y}$ is an additional matrix variable of proper dimensions.

Example 3.10. *Let us consider again the problem of the $\mathcal{H}_\infty$ norm computation for a 2-D discrete system (see Example 3.8). During formulating this problem as the GEVP, a semi-definite constraint appeared. However, this constraint can be replaced by another one to obtain the LMI problem formulation as it has been presented above.*

According to this replacement, the problem of finding of minimal $\mathcal{H}_\infty$ norm bound γ can be formulated as GEVP and has the following form

$$\begin{aligned} &\min_{\boldsymbol{P}\succ 0, \boldsymbol{Q}\succ 0, \boldsymbol{Y}\succ 0} \quad \mu \\ &\text{subject to} \begin{cases} \boldsymbol{Y} \prec \mu \boldsymbol{I} \\ \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A}+\boldsymbol{R}-\boldsymbol{S}+\boldsymbol{C}^T\boldsymbol{C} & \widehat{\boldsymbol{C}}^T\widehat{\boldsymbol{D}}+\boldsymbol{A}^T\boldsymbol{P}\boldsymbol{B} \\ \boldsymbol{B}^T\boldsymbol{P}\boldsymbol{A}+\widehat{\boldsymbol{D}}^T\widehat{\boldsymbol{C}} & \widehat{\boldsymbol{D}}^T\widehat{\boldsymbol{D}}+\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{B} \end{bmatrix} \prec \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y} \end{bmatrix} \end{cases} \end{aligned} \tag{3.18}$$

where $\mu = \gamma^2$ and $\boldsymbol{Y}$ is additional variable of proper dimension. In case of the system (3.13) the result is $\gamma = 10.2002$ (with the accuracy $\epsilon \leq 10^{-2}$).

3.4. Analytic solution of the LMI problem

It should be pointed out that in some special cases, the control problems formulated in terms of LMI (e.g. the stability problem) have an analytic solution which can be computed by direct and more effective methods. This fact can be used to check or to validate the numerical results.

It can be shown that the LMI (3.1) is equivalent to n polynomial inequalities. To see this, consider the well-known result in matrix theory (Golub and Loan, 1996) that given real symmetric matrix $\boldsymbol{F}(x) \in \mathbb{R}^{n\times n}$ is positive definite if, and only if, all of its principal minors $m_i(x)$ are positive. This means that the principal minors are multivariate polynomials of indeterminates x_i i.e.

$$
\begin{aligned}
m_1(x) =& F(x)_{11} = F_{0_{11}} + \sum_{i=1}^{n} x_i F_{i_{11}} \\
m_2(x) =& \det\left(\begin{bmatrix} F(x)_{11} & F(x)_{12} \\ F(x)_{21} & F(x)_{22} \end{bmatrix}\right) = \left(F_{0_{11}} + \sum_{i=1}^{n} x_i F_{i_{11}}\right)\left(F_{0_{22}} + \sum_{i=1}^{n} x_i F_{i_{22}}\right) \\
& - \left(F_{0_{21}} + \sum_{i=1}^{n} x_i F_{i_{21}}\right)\left(F_{0_{12}} + \sum_{i=1}^{n} x_i F_{i_{12}}\right) \\
m_k(x) =& \det\left(\begin{bmatrix} F(x)_{11} & \cdots & F(x)_{1k} \\ \vdots & \ddots & \vdots \\ F(x)_{k1} & \cdots & F(x)_{kk} \end{bmatrix}\right) \\
m_n(x) =& \det(\boldsymbol{F}(x)) = \det\left(\begin{bmatrix} F(x)_{11} & \cdots & F(x)_{1n} \\ \vdots & \ddots & \vdots \\ F(x)_{n1} & \cdots & F(x)_{kn} \end{bmatrix}\right)
\end{aligned}
$$

where $F(x)_{kl}$ denotes the element on k-th row and l-th column of $\boldsymbol{F}(x)$.

Example 3.11. *Consider again the problem of finding a block-diagonal matrix $\boldsymbol{P} \succ 0$ ($\boldsymbol{P} = \operatorname{diag}(\boldsymbol{P}_h, \boldsymbol{P}_v)$) such that the following LMI*

$$\boldsymbol{A}^T \boldsymbol{P} \boldsymbol{A} - \boldsymbol{P} \prec 0 \tag{3.19}$$

or

$$-\boldsymbol{A}^T \boldsymbol{P} \boldsymbol{A} + \boldsymbol{P} \succ 0 \tag{3.20}$$

is satisfied. Since $\boldsymbol{P} = \operatorname{diag}(x_1, x_2)$ and the matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} 0.4942 & 0.5706 \\ 0.1586 & 0.4662 \end{bmatrix} \tag{3.21}$$

then the solution of the LMI (3.20) is equivalent to the solution of the set of inequalities

$$
\begin{aligned}
m_1(x) =& - x_1(a_{11}^2 - 1) - x_2 a_{21}^2 = 0.75576636 x_1 - 0.02515396 x_2 > 0 \\
m_2(x) =& - x_1(a_{11}^2 - 1) - x_2 a_{21}^2 = 0.32558436 x_1 + 0.78265756 x_2 > 0 \\
m_3(x) =& \left(-x_1(a_{11}^2 - 1) - x_2 a_{21}^2\right)\left(-x_1 a_{12}^2 - x_2(a_{22}^2 - 1)\right) \\
& -(-x_1 a_{12} a_{11} - x_2 a_{22} a_{21})(-x_1 a_{12} a_{11} - x_2 a_{22} a_{21}) \\
=& - 0.32558436 x_1^2 + 0.5579956166 x_1 x_2 - 0.02515396 x_2^2 > 0
\end{aligned} \tag{3.22}
$$

with

$$x_1 > 0 \text{ and } x_2 > 0 \tag{3.23}$$

On the other hand, recall that the LMI (3.1) is a convex set in $\mathbb{R}^n$ defined as

$$\mathcal{F} = \left\{ x \in \mathbb{R}^n : \boldsymbol{F}(x) = \boldsymbol{F_0} + \sum_{i=1}^{n} x_i \boldsymbol{F_i} \succeq 0 \right\}$$

which can be described in terms of principal minors as

$$\mathcal{F} = \{x \in \mathbb{R}^n : m_i(x) \geq 0,\ i = 1, \ldots, n\}$$

Hence the inequalities (3.22) and (3.23) describe the convex set which is depicted in Fig. 3.4 as the shaded area. Additionally, the constraints $x_1 < 2$ *and* $x_2 < 2$ *have been imposed to bound the plot. This plot has been created with* YALMIP *by running the following sequence of commands*

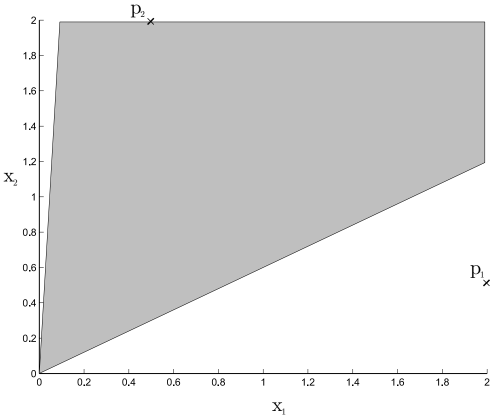


Fig. 3.4. Feasible set plot.

```
A =[ 0.4942    0.5706; 0.1586    0.4662]% the matrix A
 x = sdpvar(2,1); % two decision variable: x(1) and x(2)
 F = set(-(A(1,1)^2)*x(1)-x(2)*A(2,1)^2+x(1)>0)
 F = F+set(-(A(1,2)^2)*x(1)-x(2)*A(2,2)^2+x(2)>0)
 F = F+set(x(1)>0)
 F = F+set(x(2)>0)
 F = F+set(x(1)<2)
 F = F+set(x(2)<2)

 F = F+set([-(A(1,1)^2)*x(1)-x(2)*A(2,1)^2+x(1),...
          -A(1,1)*A(1,2)*x(1)-A(2,1)*A(2,2)*x(2);...
          -A(1,1)*A(1,2)*x(1)-A(2,1)*A(2,2)*x(2),...
           -(A(1,2)^2)*x(1)-x(2)*A(2,2)^2+x(2)]>0)
 plot(F) % plot the feasible set
```

To validate the result, computations for two points $p_1 = (x_1, x_2) = (2, 0.5)$ *and* $p_2 = (x_1, x_2) = (0.5, 2)$ *will be provided. First consider the point* p_1*. In this case, the matrix below is obtained from the LMI (3.19).*

$$\boldsymbol{R} = \boldsymbol{A}^T \boldsymbol{P} \boldsymbol{A} - \boldsymbol{P} = \begin{bmatrix} -1.4990 & 0.6010 \\ 0.6010 & 0.2598 \end{bmatrix}$$

Because eigenvalues of the matrix $\boldsymbol{R}$ *are* $\lambda_1 = -1.6847$ *and* $\lambda_2 = 0.4456$*, it is clear that* p_1 *is not the solution of the considered LMI (see Fig. 3.4 that* p_1 *does not lie* *inside the feasible set). Taking p_2 into computation yields*

$$\boldsymbol{R} = \begin{bmatrix} -0.3276 & 0.2889 \\ 0.2889 & -1.4025 \end{bmatrix}$$

which is negative defined (its eigenvalues are $\lambda_1 = -1.4752$ and $\lambda_2 = -0.2549$). On the other hand, evaluating the principal minors (3.22) yields

Table 3.3. Values of the principal minors.

	p_1	p_2
$m_1(x)$	0.3759836866	0.3759836866
$m_2(x)$	-0.259839940	1.402522940
$m_3(x)$	1.498955740	0.327575260

These results clearly show that in the case of the point (p_1 not all principal minors are positive, hence we conclude again that this point does not solve the LMI (3.19) with (3.21).

3.5. Methods to reformulate hard problems into LMIs

This section is devoted to methods which are useful in cases when it is necessary to obtain the LMI form the non-LMI formulation i.e. when a matrix inequality is not linear in respect to its parameters (e.g. the BMI form).

Before the main methods will be described, an important fact from the matrix theory is presented. This will be helpful to many transformations required by the proofs. That is, if some matrix $\boldsymbol{F}(x)$ is positive defined than $z^T\boldsymbol{F}(x)z \succ 0$, $\forall z \neq 0$, $z \in \mathbb{R}^n$. Assume now that $z = \boldsymbol{M}y$ where $\boldsymbol{M}$ is any given nonsingular matrix, hence

$$z^T\boldsymbol{F}(x)z \succ 0$$

implies that

$$y^T\boldsymbol{M}^T\boldsymbol{F}(x)\boldsymbol{M}y \succ 0$$

This means that some rearrangements of the matrix elements do not change the feasible set of LMIs. For example, if the following LMI is feasible

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \prec 0$$

then immediately the following LMI is feasible too

$$\begin{bmatrix} \boldsymbol{D} & \boldsymbol{C} \\ \boldsymbol{B} & \boldsymbol{A} \end{bmatrix} \prec 0$$

where

$$\begin{bmatrix} \boldsymbol{D} & \boldsymbol{C} \\ \boldsymbol{B} & \boldsymbol{A} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$$

3.5.1. Schur complement formula

Quadratic but convex inequality can be converted into the LMI form using Schur complement formula given by the following Lemma.

Lemma 7. *(Boyd et al., 1994) Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ be symmetric matrices and $\boldsymbol{A} \succ 0$ then*

$$\boldsymbol{C} + \boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{B} \prec 0$$

if and only if

$$\boldsymbol{U} = \begin{bmatrix} -\boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix} \prec 0 \quad \text{or, equivalently,} \quad \boldsymbol{U} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{B}^T \\ \boldsymbol{B} & -\boldsymbol{A} \end{bmatrix} \prec 0$$

The matrix $\boldsymbol{C}+\boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{B}$ is called the Schur complement of $\boldsymbol{A}$ in $\boldsymbol{U}$. The identical result holds for a positive defined case.

Example 3.12. *Consider a controller design for discrete LRPs (2.16). It can be shown that the following LMI gives sufficient condition for stability along the pass*

$$(\boldsymbol{\Phi} + \boldsymbol{R}\boldsymbol{K})^T\boldsymbol{W}(\boldsymbol{\Phi} + \boldsymbol{R}\boldsymbol{K}) - \boldsymbol{W} \prec 0 \tag{3.24}$$

where $\boldsymbol{W} \succ 0$ is block-diagonal matrix variable, $\boldsymbol{\Phi}$ and $\boldsymbol{R}$ are given matrices identified in process state-space model (2.16) as

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix}, \ \boldsymbol{R} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} \end{bmatrix}$$

and

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \\ \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix}$$

is the matrix to be found. Applying the Schur complement formula to (3.24) yields

$$\begin{bmatrix} -\boldsymbol{W}^{-1} & \boldsymbol{\Phi} + \boldsymbol{R}\boldsymbol{K} \\ \boldsymbol{\Phi}^T + \boldsymbol{K}^T\boldsymbol{R}^T & -\boldsymbol{W} \end{bmatrix} \prec 0$$

The above form is still nonlinear due to the occurrence of terms $\boldsymbol{W}^{-1}$ and $\boldsymbol{W}$ (hence it can be stated in terms of BMI). To overcome this problem, introduce the substitution $\boldsymbol{P} = \boldsymbol{W}^{-1}$ and then multiply the result from the left and the right by $\operatorname{diag}(\boldsymbol{I}, \boldsymbol{P})$ *to obtain*

$$\begin{bmatrix} -\boldsymbol{P} & \boldsymbol{\Phi}\boldsymbol{P} + \boldsymbol{R}\boldsymbol{N} \\ \boldsymbol{P}\boldsymbol{\Phi}^T + \boldsymbol{N}^T\boldsymbol{R}^T & -\boldsymbol{P} \end{bmatrix} \prec 0 \tag{3.25}$$

where $\boldsymbol{N} = \boldsymbol{K}\boldsymbol{P}$. Now, it is straightforward to see that (3.25) is the feasibility problem (see Definition 3.1), which is numerically solvable.

3.5.2. Elimination of a norm bounded matrix

In robustness analysis, we often encounter the following terms

$$\boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E} + \boldsymbol{E}^T\boldsymbol{\mathcal{F}}^T\boldsymbol{H}^T \tag{3.26}$$

where $\boldsymbol{H}$, $\boldsymbol{E}$ are known real matrices of appropriate dimensions, and the matrix $\boldsymbol{\mathcal{F}}$ represents parameter uncertainties which satisfies

$$\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}} \preceq \boldsymbol{I} \text{ or equivalently } \|\boldsymbol{\mathcal{F}}\| \le 1$$

Inequalities which consist of (3.26) can be transformed into the LMI with the following Lemma

Lemma 8. *Let $\boldsymbol{H}$, $\boldsymbol{E}$ be given real matrices of appropriate dimensions and $\boldsymbol{\mathcal{F}}$ satisfy $\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}} \preceq \boldsymbol{I}$. Then for any $\epsilon > 0$ the following holds*

$$\boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E} + \boldsymbol{E}^T\boldsymbol{\mathcal{F}}^T\boldsymbol{H}^T \preceq \epsilon\boldsymbol{H}\boldsymbol{H}^T + \frac{1}{\epsilon}\boldsymbol{E}^T\boldsymbol{E}$$

Proof. *Since it is true that*

$$\left(\epsilon^{\frac{1}{2}}\boldsymbol{H}^T - \epsilon^{-\frac{1}{2}}\boldsymbol{\mathcal{F}}\boldsymbol{E}\right)^T \left(\epsilon^{\frac{1}{2}}\boldsymbol{H}^T - \epsilon^{-\frac{1}{2}}\boldsymbol{\mathcal{F}}\boldsymbol{E}\right) \succeq 0$$

then expansion of the above yields

$$\epsilon^{-1}\boldsymbol{E}^T\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}}\boldsymbol{E} + \epsilon\boldsymbol{H}\boldsymbol{H}^T \succeq \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E} + \boldsymbol{E}^T\boldsymbol{\mathcal{F}}^T\boldsymbol{H}^T$$

Next, observe that

$$\|\boldsymbol{\mathcal{F}}\| \le 1 \Leftrightarrow \lambda_{\max}(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}}) \le 1 \Leftrightarrow \boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}} \preceq \boldsymbol{I}$$

hence

$$\epsilon\boldsymbol{H}\boldsymbol{H}^T + \frac{1}{\epsilon}\boldsymbol{E}^T\boldsymbol{E} \succeq \epsilon^{-1}\boldsymbol{E}^T\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}}\boldsymbol{E} + \epsilon\boldsymbol{H}\boldsymbol{H}^T \succeq \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E} + \boldsymbol{E}^T\boldsymbol{\mathcal{F}}^T\boldsymbol{H}^T$$

and the proof is complete ■

Note that some part of the proof can be found in (Oliveira, 2002).

3.5.3. Elimination of variables

For certain specific matrix inequalities, if is often possible to eliminate some of the matrix variables.

Lemma 9. *(Gahinet and Apkarian, 1994; Iwasaki and Skelton, 1994) Let $\boldsymbol{\Psi} \in \mathbb{R}^{q\times q}$ be a symmetric matrix and $\boldsymbol{P} \in \mathbb{R}^{r\times q}$ and $\boldsymbol{Q} \in \mathbb{R}^{s\times q}$ be real matrices then there exists a matrix $\boldsymbol{\Theta} \in \mathbb{R}^{r\times s}$ such that*

$$\boldsymbol{\Psi} + \boldsymbol{P}^T\boldsymbol{\Theta}^T\boldsymbol{Q} + \boldsymbol{Q}^T\boldsymbol{\Theta}\boldsymbol{P} \prec 0 \tag{3.27}$$

if and only if the inequalities

$$\boldsymbol{\mathcal{W}}_P^T \Psi \boldsymbol{\mathcal{W}}_P \prec 0 \text{ and } \boldsymbol{\mathcal{W}}_Q^T \Psi \boldsymbol{\mathcal{W}}_Q \prec 0$$

both hold, where $\boldsymbol{\mathcal{W}}_P$ *and* $\boldsymbol{\mathcal{W}}_Q$ *are full rank matrices satisfying* $\mathrm{Im}(\boldsymbol{\mathcal{W}}_P) = \ker(\boldsymbol{P})$ *and* $\mathrm{Im}(\boldsymbol{\mathcal{W}}_Q) = \ker(\boldsymbol{Q})$

This is a key lemma in stabilisation problems where inequalities of the form (3.27) appear. However, it can also be used to eliminate variables from already formulated LMI. Since some variables can be eliminated, the computation burden can be reduced greatly. To see this, the following example is provided.

Example 3.13. *Consider again the stabilisation problem presented in Example 3.12. The right-hand term in (3.25) can be rewritten as*

$$\begin{bmatrix} -\boldsymbol{P} & \boldsymbol{\Phi P} + \boldsymbol{RN} \\ \boldsymbol{P\Phi}^T + \boldsymbol{N}^T\boldsymbol{R}^T & -\boldsymbol{P} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{P} & \boldsymbol{\Phi P} \\ \boldsymbol{P\Phi}^T & -\boldsymbol{P} \end{bmatrix} + \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{N} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{N}^T \begin{bmatrix} \boldsymbol{R}^T & \boldsymbol{0} \end{bmatrix}$$

Using Lemma 9, we obtain

$$\boldsymbol{\mathcal{W}}_R^T \begin{bmatrix} -\boldsymbol{P} & \boldsymbol{\Phi P} \\ \boldsymbol{P\Phi}^T & -\boldsymbol{P} \end{bmatrix} \boldsymbol{\mathcal{W}}_R^T \prec 0, \quad \boldsymbol{\mathcal{W}}_S^T \begin{bmatrix} -\boldsymbol{P} & \boldsymbol{\Phi P} \\ \boldsymbol{P\Phi}^T & -\boldsymbol{P} \end{bmatrix} \boldsymbol{\mathcal{W}}_S^T \prec 0$$

where $\boldsymbol{\mathcal{W}}_R = \mathrm{diag}(\ker(\boldsymbol{R}), \boldsymbol{I})$ *and* $\boldsymbol{\mathcal{W}}_S = \mathrm{diag}(\boldsymbol{I}, \boldsymbol{0})$. *These two LMI conditions can be checked with less computation burden than the LMI condition provided in Example 3.12. Illustrative computations have been performed for processes of prescribed order (n) and the results are listed in Table 3.4. Note that all computations have been performed with* Lmi Control Toolbox *1.0.8 under* Matlab *6.5. The* Matlab- *les have been run on a PC with AMD Duron 600 MHz CPU and 128MB RAM.*

Table 3.4. Execution time comparison.

n	Example 3.12 (CPU time in sec.)	This Example (CPU time in sec.)
6	0.11	0.06
8	0.22	0.11
12	1.15	0.6
15	19.06	1.76
20	73.44	7.91

3.6. Concluding remarks

In this chapter, LMI methods which have become a standard in system theory, have been described in detail. It has been done by providing the mathematical theory required to understand what LMIs are and to manipulate them. It is shown that the LMI approach is very attractive due to its numerical advantages and relative simplicity. These numerical advantages lie in the fact that computations required to solve LMIs maintain a reasonable computational cost i.e. the computational cost of solving LMIs is a polynomial function of the number of decision variables. Hence it is clear that a problem formulated in terms of LMI belongs to the class of $\mathcal{P}$-problems.

Moreover, LMIs provide an alternate problem formulation in 1-D system theory that avoids direct manipulation on system poles. This makes LMIs especially useful in the solving of n-D system analysis and synthesis problems, because computations over system poles are omitted.

Chapter 4

ROBUSTNESS ANALYSIS WITH LMI METHODS

It turns out that LMI methods offer the chance to approximate many $\mathcal{NP}$-hard problems via polynomial-time algorithms and therefore, these methods make it possible to find a solution for such non-trivial problems. One of the class of problems for which LMI methods are well suited, are control problems with uncertain data. Indeed, there are many examples of control problems with uncertain data which have been effectively solved using LMI methods. These solution are frequently obtained under the assumption that a system matrix entries range in a given convex set and hence such a set can be described in terms of LMI (El Ghaoui and Niculescu, 1999).

A fundamental point to note is that most of the known results are only presented for 1-D systems. Therefore, due to the lack of results in robust analysis and synthesis of LRPs, this chapter provides them. Resulting conditions are formulated in terms of LMI and hence, in the sense of computational complexity, this approach shows that robust analysis and synthesis problems for LRPs and n-D systems in general, can be reduced to polynomial-time problems.

4.1. Computed-aided methods for robustness problems

It is known that most robustness problems in both 1-D and n-D system theory have been proven to be $\mathcal{NP}$-hard (Blondel and Tsitsiklis, 2000b). One of the ways to overcome the computational problems which have arisen in robustness analysis and synthesis is to use simulation and numerical tools to find an upper bound on parameter values for which a system remains stable.

Using this approach, the robustness of LRP or n-D system with uncertain parameters can be determined with many simulations of LRP or n-D system performed under various scenarios, i.e. with many different parameter values. For each run, the simulation results are noted and the influence of parameter values are studied. An important advantage of such a procedure is that it is easy to perform and an engineer who is interested in an LRP or n-D system robustness is not required to be an expert in advanced mathematics or n-D system theory. Moreover, this technique can be applied for a very wide variety of n-D systems, including n-D systems with delays. Obviously, values of an uncertain parameter belong to a set which usually has an infinite number of elements, therefore we cannot perform all simulations in finite time. Even though we are interested in performing a limited number of simulations, we have to keep in mind that the resulting conclusion is only restricted to used parameter values. Further, we do not know for which parameter values the simulations have to be performed.

To reduce some of these drawbacks the methods based on LMI problem formulation can be used. However, the main difficulty behind the direct application of LMI methods to solve a robust design problem is the fact that the stability region (a set of all stable matrices) is generally non-convex. This means that the problem of interest can be generally stated in terms of BMI, which is extremely difficult to solve by existing global optimization approaches. To overcome this problem, a convex approximation of non-convex stability region is used. This immediately results in a sufficiency of obtained LMI conditions, but it is often about the best we can do.

To proceed, robustness problems in LRP (2-D) system theory e.g. robust stability and robust stabilisation problems will be reformulated in an LMI framework with the help of convex analysis and matrix theory. In what follows, uncertain parameters are assumed to vary in a prescribed convex set, therefore such a set can be described in terms of LMI.

Suppose now that uncertainty is modelled with a norm-bounded model (2.39)-(2.40). In this case, the inequality (2.40) represents a convex set. To see this, apply the Schur complement formula to obtain

$$\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}} \preceq \boldsymbol{I} \quad \Leftrightarrow \quad \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\mathcal{F}} \\ \boldsymbol{\mathcal{F}}^T & \boldsymbol{I} \end{bmatrix} \succeq 0 \tag{4.1}$$

It is illustrated by the following example.

Example 4.1. *Since the matrix $\boldsymbol{\mathcal{F}}$ is assumed to be dimension 2×1, then the convex set represented by (4.1) can be depicted on a two-dimensional plane. Consider the following* YALMIP *program*

```
x = sdpvar(2,1);                    % two variables
F = set([eye(2), x; x', 1]>0);      % the constraint (4.1)
plot(F);
```

which generates the Fig. 4.1. It is straightforward to see that the set depicted in this figure is the convex set.

It was mentioned that the important property of LMIs is that they form a convex constraint on the decision variables vector x. This means that any convex combination of solutions taken from a feasible set of LMIs, are also a solution. This also means that in case of uncertainty modelled with a polytopic model (see the equation (2.41)), we only need to find a solution for all vertices of the polytop to obtain a solution for all elements of the uncertainty set. The main advantage associated with this fact is that we need to compute a solution for a finite number of LMIs which clearly involves a finite amount of computation.

Affine parameter-dependent matrices (2.42) can be easily transformed to the polytopic form (2.41). It stems from the fact that parameters change their values

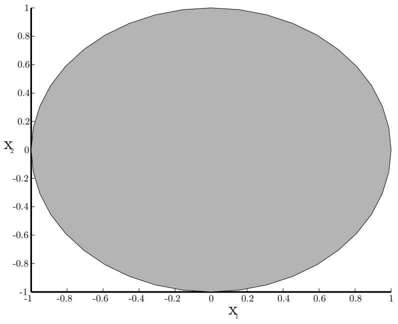


Fig. 4.1 . The set represented by the inequality (4.1).

in the parameter box with 2^k corners (see Fig. 2.8), where k denotes the number of parameters. It is clear that $\boldsymbol{M}(p)$ is an affine function in $p = (p_1, p_2, \ldots, p_k)$, thus it maps these corners to the polytope of vertices. In this case each vertex can be determined $\forall p \in \Delta_0$ with the formula below

$$\boldsymbol{M}_i = \boldsymbol{M}_0 + p_1\boldsymbol{M}_1 + \ldots + p_k\boldsymbol{M}_k \tag{4.2}$$

where $i = 1, .., 2^k$. The graphical illustration of this transformation (for the parameters p_1 and p_2) is depicted in Fig. 4.2. Because of such a transformation, analysis and synthesis problems with uncertain data, modelled with the affine model of uncertainty can be considered as polynomial-time solvable.

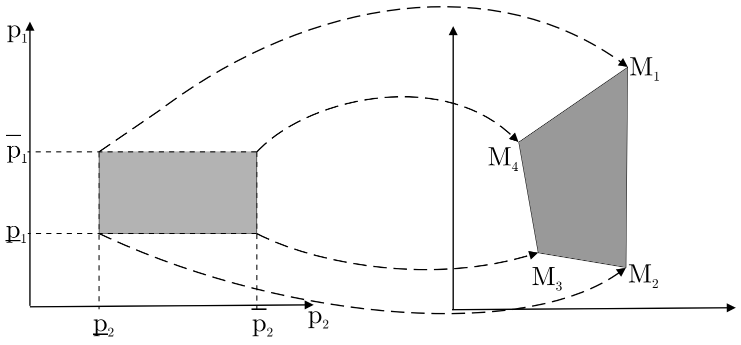


Fig. 4.2 . Transformation from the affine model to the polytopic model.

4.2. Robust stability and stabilisation of differential LRPs

One key area for which no results are currently available is stability and stabilisation of differential LRPs in the presence of uncertainties in the model structure. The presence of these uncertainties in the matrices, which define the state-space model of the process under consideration, requires conditions that ensures the process to be robustly stable for all admissible uncertainties. Robust stability conditions can be further employed to solve the robust stabilisation problem.

4.2.1. Robust stability

4.2.1.1. Norm-bounded model of uncertainty

Consider uncertain differential LRPs described by the following state-space model over $0 \leq t \leq \alpha$, $k \geq 0$

$$\begin{aligned} \dot{x}_{k+1}(t) =& (\boldsymbol{A} + \Delta\boldsymbol{A})x_{k+1}(t) + (\boldsymbol{B}_0 + \Delta\boldsymbol{B}_0)y_k(t) + (\boldsymbol{B} + \Delta\boldsymbol{B})u_{k+1}(t) \\ y_{k+1}(t) =& (\boldsymbol{C} + \Delta\boldsymbol{C})x_{k+1}(t) + (\boldsymbol{D}_0 + \Delta\boldsymbol{D}_0)y_k(t) + (\boldsymbol{D} + \Delta\boldsymbol{D})u_{k+1}(t) \end{aligned} \tag{4.3}$$

The matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{B}_0$, $\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{D}_0$ define the nominal model (2.14) and $\Delta\boldsymbol{A}$, $\Delta\boldsymbol{B}$, $\Delta\boldsymbol{B}_0$, $\Delta\boldsymbol{C}$, $\Delta\boldsymbol{D}$, $\Delta\boldsymbol{D}_0$ represent admissible uncertainties which are assumed to be of the form

$$\begin{bmatrix} \Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0 & \Delta\boldsymbol{B} \\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 & \Delta\boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} \boldsymbol{\mathcal{F}} \begin{bmatrix} \boldsymbol{E}_1 & \boldsymbol{E}_2 & \boldsymbol{E}_3 \end{bmatrix} \tag{4.4}$$

In this last equation, $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ are known constant matrices of compatible dimensions, and $\boldsymbol{\mathcal{F}}$ is an unknown matrix with constant entries which satisfies

$$\boldsymbol{\mathcal{F}}^T \boldsymbol{\mathcal{F}} \preceq \boldsymbol{I} \tag{4.5}$$

For a stability goal, consider the state-space model (4.3) with no control inputs i.e. $u_{k+1}(t) = 0$, then by defining the following matrices

$$\widehat{\boldsymbol{A}}_1 = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \ \widehat{\boldsymbol{A}}_2 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix} \tag{4.6}$$

and

$$\Delta\widehat{\boldsymbol{A}}_1 = \begin{bmatrix} \Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \ \Delta\widehat{\boldsymbol{A}}_2 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 \end{bmatrix} \tag{4.7}$$

the process state-space model (4.3) can be rewritten as

$$\begin{aligned} \begin{bmatrix} \dot{x}_{k+1}(t) \\ y_{k+1}(t) \end{bmatrix} &= \left(\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix} + \begin{bmatrix} \Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0 \\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 \end{bmatrix} \right) \begin{bmatrix} x_{k+1}(t) \\ y_k(t) \end{bmatrix} \\ &= \left(\widehat{\boldsymbol{A}}_1 + \Delta\widehat{\boldsymbol{A}}_1 \right) \xi(k,t) + \left(\widehat{\boldsymbol{A}}_2 + \Delta\widehat{\boldsymbol{A}}_2 \right) \xi(k,t) \end{aligned}$$

where

$$\xi(k,t) = \begin{bmatrix} x_{k+1}(t) \\ y_k(t) \end{bmatrix} \tag{4.8}$$

With this notation, we have the following sufficient condition for stability along the pass in terms of LMI feasibility problem.

Theorem 4.1. *A differential LRP described by (4.3) is stable along the pass for all admissible uncertainties if there exist matrices $\boldsymbol{P}_1 \succ 0$, $\boldsymbol{P}_2 \succ 0$ and a scalar $\epsilon > 0$ such that*

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A}+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \boldsymbol{P}_1\boldsymbol{B}_0 & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & -\boldsymbol{P}_2+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{0} & \boldsymbol{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \quad (4.9)$$

Proof. *Let us choose the Lyapunov functional candidate as that defined in (2.36). Since*

$$\dot{V}_1(k,t) = \dot{x}_{k+1}^T(t)\boldsymbol{P}_1 x_{k+1}(t) + x_{k+1}^T\boldsymbol{P}_1\dot{x}_{k+1}(t)$$

and

$$\Delta V_2(k,t) = y_{k+1}^T(t)\boldsymbol{P}_2 y_{k+1}(t) - y_k^T(t)\boldsymbol{P}_2 y_k(t)$$

then the associated increment for (2.36) is

$$\begin{aligned}\Delta V(k,t) =&\dot{V}_1(k,t) + \Delta V_2(k,t) \\ =&\dot{x}_{k+1}^T(t)\boldsymbol{P}_1 x_{k+1}(t)+x_{k+1}^T(t)\boldsymbol{P}_1\dot{x}_{k+1}(t) \\ &+y_{k+1}^T(t)\boldsymbol{P}_2 y_{k+1}(t)-y_k^T(t)\boldsymbol{P}_2 y_k(t)\end{aligned} \quad (4.10)$$

which together with notation introduced in (4.6), (4.7) and (4.8) gives

$$\begin{aligned}\Delta V(k,t)=&\xi^T(k,t)\Big((\widehat{\boldsymbol{A}}_1+\Delta\widehat{\boldsymbol{A}}_1)^T\boldsymbol{P}+\boldsymbol{P}(\widehat{\boldsymbol{A}}_1+\Delta\widehat{\boldsymbol{A}}_1) \\ &+(\widehat{\boldsymbol{A}}_2+\Delta\widehat{\boldsymbol{A}}_2)\boldsymbol{R}(\widehat{\boldsymbol{A}}_2+\Delta\widehat{\boldsymbol{A}}_2)-\boldsymbol{R}\Big)\xi(k,t)\end{aligned} \quad (4.11)$$

where

$$\boldsymbol{P} = \mathrm{diag}(\boldsymbol{P}_1, \boldsymbol{0}), \quad \boldsymbol{R} = \mathrm{diag}(\boldsymbol{0}, \boldsymbol{P}_2) \quad (4.12)$$

Hence stability along the pass holds if $\Delta V(k,t) < 0$ for $\forall\xi(k,t) \neq 0$, and a sufficient condition for this is

$$(\widehat{\boldsymbol{A}}_1+\Delta\widehat{\boldsymbol{A}}_1)^T\boldsymbol{P}+\boldsymbol{P}(\widehat{\boldsymbol{A}}_1+\Delta\widehat{\boldsymbol{A}}_1) +(\widehat{\boldsymbol{A}}_2+\Delta\widehat{\boldsymbol{A}}_2)\boldsymbol{S}(\widehat{\boldsymbol{A}}_2+\Delta\widehat{\boldsymbol{A}}_2)-\boldsymbol{R} \prec 0$$

where $\boldsymbol{S} = \mathrm{diag}(\boldsymbol{P}_3, \boldsymbol{P}_2)$, and $\boldsymbol{P}_3 \succ 0$ is any given matrix of appropriate dimension. Next, an obvious application of the Schur complement formula yields

$$\begin{bmatrix} -\boldsymbol{S} & \boldsymbol{S}\widehat{\boldsymbol{A}}_2 \\ \widehat{\boldsymbol{A}}_2^T\boldsymbol{S} & \widehat{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\widehat{\boldsymbol{A}}_1-\boldsymbol{R} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{S}\Delta\widehat{\boldsymbol{A}}_2 \\ \Delta\widehat{\boldsymbol{A}}_2^T\boldsymbol{S} & \Delta\widehat{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\Delta\widehat{\boldsymbol{A}}_1 \end{bmatrix} \prec 0$$

or, equivalently (the block $-\boldsymbol{P}_3$ is negative defined therefore it can be removed),

$$\begin{aligned}&\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A} & \boldsymbol{P}_1\boldsymbol{B}_0 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & -\boldsymbol{P}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \boldsymbol{0} & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{F}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2 \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E}_1^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{F}}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{F}}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{0} \end{bmatrix} \prec 0\end{aligned}$$

Based on the result of Lemma 8, we obtain

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A}+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \boldsymbol{P}_1\boldsymbol{B}_0 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & -\boldsymbol{P}_2+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 \end{bmatrix}$$

$$+\epsilon^{-1}\begin{bmatrix} \mathbf{0} & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \mathbf{0} & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \mathbf{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \mathbf{0} \end{bmatrix}\prec 0$$

Finally, using the Schur complement formula, we find that the last inequality is equivalent to the LMI (4.9). This concludes the proof. ■

Remark 4.1. *If there are no uncertain matrices in the process model (4.3) then the LMI (4.9) reduces into LMI for nominal model - see (Gałkowski et al., 2003c) for details.*

4.2.1.2. Polytopic model of uncertainty

Here it is assumed that uncertainty of the differential fraction of an uncertain differential LRP (4.3) has a polytopic character, i.e. all possible choices for the matrices which define the current pass state dynamics in (4.3) can be expressed as

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \end{bmatrix} \in \mathrm{Co}\left(\begin{bmatrix} \boldsymbol{A}^i & \boldsymbol{B}_0^i \end{bmatrix}\right) \tag{4.13}$$

where $i = 1, 2, \ldots, h$ and

$$\mathrm{Co}\left(\begin{bmatrix} \boldsymbol{A}^i & \boldsymbol{B}_0^i \end{bmatrix}\right) := \left\{\boldsymbol{X} : \boldsymbol{X} = \sum_{i=1}^{h} \alpha_i \begin{bmatrix} \boldsymbol{A}^i & \boldsymbol{B}_0^i \end{bmatrix}, \quad \alpha_i \geq 0, \quad \sum_{i=1}^{h} \alpha_i = 1\right\} \tag{4.14}$$

For the current pass profile updating equation in (4.3) a standard norm-bound on the perturbations is assumed, i.e. the second equation entry in (4.3) takes the form

$$y_{k+1}(t) = (\boldsymbol{C}+\Delta\boldsymbol{C})x_{k+1}(t) + (\boldsymbol{D}_0+\Delta\boldsymbol{D}_0)y_k(t) \tag{4.15}$$

where

$$\begin{bmatrix} \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 \end{bmatrix} = \boldsymbol{H}_2\boldsymbol{\mathcal{F}}\begin{bmatrix} \boldsymbol{E}_1 & \boldsymbol{E}_2 \end{bmatrix}$$

and the matrix $\boldsymbol{\mathcal{F}}$ satisfies (4.5). The following result gives an LMI-based sufficient condition for stability along the pass.

Theorem 4.2. *A differential LRP of the form described by (4.3), with uncertainty structure modelled by (4.13)-(4.14) and (4.15) is stable along the pass for all admissible uncertainties if there exist matrices $\boldsymbol{P}_1 \succ 0$, $\boldsymbol{P}_2 \succ 0$ and a scalar $\epsilon > 0$, such that*

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^{iT}\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A}^i+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \boldsymbol{P}_1\boldsymbol{B}_0^i+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_2 & \mathbf{0} \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^{iT}\boldsymbol{P}_1+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_1 & -\boldsymbol{P}_2+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 & \mathbf{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \tag{4.16}$$

Proof. *Taking account of the uncertainty in the considered process model, we get*

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^{iT}\boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A}^i & \boldsymbol{P}_1\boldsymbol{B}_0^i \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^{iT}\boldsymbol{P}_1 & -\boldsymbol{P}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{E}_1^T \\ \boldsymbol{E}_2^T \end{bmatrix} \boldsymbol{\mathcal{F}}^T \begin{bmatrix} \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$
$$+ \begin{bmatrix} \boldsymbol{P}_2\boldsymbol{H}_2 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\mathcal{F}} \begin{bmatrix} \mathbf{0} & \boldsymbol{E}_1 & \boldsymbol{E}_2 \end{bmatrix} \prec 0$$

which in view of Lemma 8 is equivalent to

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^{iT}\boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A}^i + \epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \boldsymbol{P}_1\boldsymbol{B}_0^i + \epsilon\boldsymbol{E}_1^T\boldsymbol{E}_2 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^{iT}\boldsymbol{P}_1 + \epsilon\boldsymbol{E}_2^T\boldsymbol{E}_1 & -\boldsymbol{P}_2 + \epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 \end{bmatrix}$$
$$+ \epsilon^{-1} \begin{bmatrix} \boldsymbol{P}_2\boldsymbol{H}_2 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix} \prec 0$$

Finally, application of the Schur complement formula gives (4.16). This completes the proof. ■

Remark 4.2. *Some comments are required for Theorem 4.2. First, it should be emphasized that the result has been obtained by keeping $\boldsymbol{P}_1$ constant and independent of the index i. The main drawback associated with this fact is that the Lyapunov matrix $\boldsymbol{P}_1$ must work for all uncertain matrices (4.14). This condition can introduce a significant degree of conservativeness to the stability condition (4.16). This can be overcome by using parameter-dependent Lyapunov functions (Apkarian and Tuan, 1998, 2000). Unfortunately, there is no result for LRPs where parameter-dependent Lyapunov functions are used. Work is proceeding on applying such functions and will be reported on in due course.*

It is clear that computation can be only performed in the case when all vertices (i.e. all edge matrices $\boldsymbol{A}^i$ and $\boldsymbol{B}_0^i$) of a convex polygon are known. Unfortunately, in many practical cases they are unknown, so it is required to determine these vertices. This is best done with a numerical tool run on a computer. The example of such a tool is Geometric Bounding Toolbox (Veres, 2004) which provides computations in n-dimensional space. The main objective of this toolbox is the computation of the convex hull of a finite set of points (command `convh`) which is useful for our purposes. The simplicity of using this software package is demonstrated with the example below.

Example 4.2. *The following set of commands generates a random set of points in three-dimensional space and compute their convex hull*

```
points3D = rand(7,3)   % 7 random points in 3-D space
cxhull=convh(points3D) % determine their convex hull
```

However, in the case of systems considered here where matrix representation appears, it is necessary to perform transformation from space of real m by n matrices to mn-dimensional real vector space, to make computations with the Geometric Bounding Toolbox possible. Then, after computing a convex hull in that real vector space, return to space of matrices and determine what matrices have to be used in further computations.

4.2.1.3. Affine model of uncertainty

It was shown that the affine model of uncertainty could be easily transformed to the polytopic one where a set of vertices corresponds to extremal values of the parameter vector. Faced with this fact, the affine model of uncertainty is not considered to be present here any more. On the other hand, the affine model allows us to maximize a stability region, i.e. the largest portion of the parameter box where the stability along the pass can be established (for 1-D system case see, for example, (Gahinet *et al.*, 1995)). To proceed, define the nominal value of the parameter p_i as

$$p_{0_i} = \frac{\overline{p}_i + \underline{p}_i}{2}$$

which denotes the center of the interval $[\underline{p_i}, \overline{p_i}]$. Next, for each interval define its radius by

$$\delta_i = \frac{\overline{p}_i - \underline{p}_i}{2}$$

to yield

$$p_i \in [p_{0_i} - \mu\delta_i, p_{0_i} + \mu\delta_i]$$

where $\mu > 0$ is the unknown scalar to be selected. Maximization of μ leads to maximization of the parameter box wherein stability along the pass is guaranteed. It turns out that maximization of μ can be cast into GEVP.

4.2.2. Robust stabilisation

In practice, it can happen that an uncertain process can be unstable, hence the robust stabilisation problem has to be addressed. To proceed, the following form of the control law is used

$$u_{k+1}(t) = \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix} \begin{bmatrix} x_{k+1}(t) \\ y_k(t) \end{bmatrix} \tag{4.17}$$

where $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are appropriately dimensioned matrices to be designed. In effect, this control law uses feedback of the current state vector (which is assumed to be available for use) and 'feedforward' of the previous pass profile vector. Note that in repetitive processes, 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.

4.2.2.1. Norm-bounded model of uncertainty

Application of the control law (4.17) to (4.3) gives the following form of the closed-loop process

$$\begin{aligned}\begin{bmatrix}\dot{x}_{k+1}(t)\\ y_{k+1}(t)\end{bmatrix} =& \left(\begin{bmatrix}\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{B}_0+\boldsymbol{B}\boldsymbol{K}_2\\ \boldsymbol{C}+\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{D}_0+\boldsymbol{D}\boldsymbol{K}_2\end{bmatrix}\right.\\ &+\left.\begin{bmatrix}\Delta\boldsymbol{A}+\Delta\boldsymbol{B}\boldsymbol{K}_1 & \Delta\boldsymbol{B}_0+\Delta\boldsymbol{B}\boldsymbol{K}_2\\ \Delta\boldsymbol{C}+\Delta\boldsymbol{D}\boldsymbol{K}_1 & \Delta\boldsymbol{D}_0+\Delta\boldsymbol{D}\boldsymbol{K}_2\end{bmatrix}\right)\begin{bmatrix}x_{k+1}(t)\\ y_k(t)\end{bmatrix}\end{aligned} \tag{4.18}$$

where the admissible uncertainties are assumed to be of the form (4.4) and (4.5). In this case, the following theorem gives the condition of existing a controller that stabilises the process (4.3) for all admissible uncertainties.

Theorem 4.3. *Suppose that a differential LRP of the form described by (4.3), with uncertainty structure modelled by (4.4) and (4.5), is subjected to a control law of the form (4.17). Then the resulting closed-loop process is stable along the pass for all admissible uncertainties if there exist matrices* $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$ *and* $\boldsymbol{N}_2$ *of compatible dimensions and a scalar* $\epsilon > 0$ *such that the following LMI holds*

$$\begin{bmatrix}-\boldsymbol{W}_2+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_1^T & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_1^T & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \boldsymbol{0}\\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T\\ \boldsymbol{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 & \boldsymbol{0} & -\epsilon\boldsymbol{I}\end{bmatrix} \prec 0 \tag{4.19}$$

If the above LMI holds then the controller matrices $\boldsymbol{K}_1$ *and* $\boldsymbol{K}_2$ *are given by*

$$\begin{aligned}\boldsymbol{K}_1 &= \boldsymbol{N}_1\boldsymbol{W}_1^{-1}\\ \boldsymbol{K}_2 &= \boldsymbol{N}_2\boldsymbol{W}_2^{-1}\end{aligned} \tag{4.20}$$

respectively.

Proof. *Application of Theorem 4.1 result proves that the closed-loop process in this case is stable along the pass if*

$$\begin{bmatrix}-\boldsymbol{S} & \boldsymbol{S}\overline{\boldsymbol{A}}_2\\ \overline{\boldsymbol{A}}_2^T\boldsymbol{S} & \overline{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\overline{\boldsymbol{A}}_1-\boldsymbol{R}\end{bmatrix} + \begin{bmatrix}\boldsymbol{0} & \boldsymbol{S}\Delta\overline{\boldsymbol{A}}_2\\ \Delta\overline{\boldsymbol{A}}_2^T\boldsymbol{S} & \Delta\overline{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\Delta\overline{\boldsymbol{A}}_1\end{bmatrix} \prec 0$$

where

$$\overline{\boldsymbol{A}}_1 = \begin{bmatrix}\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{B}_0+\boldsymbol{B}\boldsymbol{K}_2\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix},\ \Delta\overline{\boldsymbol{A}}_1 = \begin{bmatrix}\Delta\boldsymbol{A}+\Delta\boldsymbol{B}\boldsymbol{K}_1 & \Delta\boldsymbol{B}_0+\Delta\boldsymbol{B}\boldsymbol{K}_2\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix},$$

$$\overline{\boldsymbol{A}}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{C}+\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{D}_0+\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix}, \Delta\overline{\boldsymbol{A}}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \Delta\boldsymbol{C}+\Delta\boldsymbol{D}\boldsymbol{K}_1 & \Delta\boldsymbol{D}_0+\Delta\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix}$$

Note that this last condition is not linear in $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$, $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$. Clearly, it is bilinear in variables $\{\boldsymbol{P}_1, \boldsymbol{P}_2\}$ and $\{\boldsymbol{K}_1, \boldsymbol{K}_2\}$ and therefore it may be considered as a BMI problem (3.4), for which effective computations cannot be performed. However, this can be reformulated as an LMI problem. To proceed, substitute $\overline{\boldsymbol{A}}_1$ and $\overline{\boldsymbol{A}}_2$ into this last expression to obtain (the block $-\boldsymbol{P}_3$ has been removed due to the fact the result does not depend on it)

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C}+\boldsymbol{P}_2\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{P}_2\boldsymbol{D}_0+\boldsymbol{P}_2\boldsymbol{D}\boldsymbol{K}_2 \\ \boldsymbol{C}^T\boldsymbol{P}_2+\boldsymbol{K}_1^T\boldsymbol{D}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{K}_1^T\boldsymbol{B}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A}+\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{P}_1\boldsymbol{B}_0+\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{K}_2 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2+\boldsymbol{K}_2^T\boldsymbol{D}^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1+\boldsymbol{K}_2^T\boldsymbol{B}^T\boldsymbol{P}_1 & -\boldsymbol{P}_2 \end{bmatrix}$$
$$+\begin{bmatrix} \mathbf{0} & \boldsymbol{P}_2\Delta\boldsymbol{C}+\boldsymbol{P}_2\Delta\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{P}_2\Delta\boldsymbol{D}_0+\boldsymbol{P}_2\Delta\boldsymbol{D}\boldsymbol{K}_2 \\ \Delta\boldsymbol{C}^T\boldsymbol{P}_2+\boldsymbol{K}_1^T\Delta\boldsymbol{D}^T\boldsymbol{P}_2 & \boldsymbol{\Omega}_1 & \boldsymbol{P}_1\Delta\boldsymbol{B}_0+\boldsymbol{P}_1\Delta\boldsymbol{B}\boldsymbol{K}_2 \\ \Delta\boldsymbol{D}_0^T\boldsymbol{P}_2+\boldsymbol{K}_2^T\Delta\boldsymbol{D}^T\boldsymbol{P}_2 & \Delta\boldsymbol{B}_0^T\boldsymbol{P}_1+\boldsymbol{K}_2^T\Delta\boldsymbol{B}^T\boldsymbol{P}_1 & \mathbf{0} \end{bmatrix} \prec 0$$

where

$$\boldsymbol{\Omega}_1 = \Delta\boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{K}_1^T\Delta\boldsymbol{B}^T\boldsymbol{P}_1+\boldsymbol{P}_1\Delta\boldsymbol{A}+\boldsymbol{P}_1\Delta\boldsymbol{B}\boldsymbol{K}_1$$

Next, pre- and post-multiply the result by $\mathrm{diag}(\boldsymbol{P}_2^{-1}, \boldsymbol{P}_1^{-1}, \boldsymbol{P}_2^{-1})$ *and make changes of variables $\boldsymbol{W}_1 = \boldsymbol{P}_1^{-1}$, $\boldsymbol{W}_2 = \boldsymbol{P}_2^{-1}$, $\boldsymbol{N}_1 = \boldsymbol{K}_1\boldsymbol{P}_1^{-1}$, $\boldsymbol{N}_2 = \boldsymbol{K}_2\boldsymbol{P}_2^{-1}$ to yield*

$$\begin{bmatrix} -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 \end{bmatrix}$$
$$+\begin{bmatrix} \mathbf{0} & \Delta\boldsymbol{C}\boldsymbol{W}_1+\Delta\boldsymbol{D}\boldsymbol{N}_1 & \Delta\boldsymbol{D}_0\boldsymbol{W}_2+\Delta\boldsymbol{D}\boldsymbol{N}_2 \\ \boldsymbol{W}_1\Delta\boldsymbol{C}^T+\boldsymbol{N}_1^T\Delta\boldsymbol{D}^T & \boldsymbol{\Omega}_2 & \Delta\boldsymbol{B}_0\boldsymbol{W}_2+\Delta\boldsymbol{B}\boldsymbol{N}_2 \\ \boldsymbol{W}_2\Delta\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\Delta\boldsymbol{D}^T & \boldsymbol{W}_2\Delta\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\Delta\boldsymbol{B}^T & \mathbf{0} \end{bmatrix} \prec 0 \tag{4.21}$$

where

$$\boldsymbol{\Omega}_2 = \boldsymbol{W}_1\Delta\boldsymbol{A}^T+\boldsymbol{N}_1^T\Delta\boldsymbol{B}^T+\Delta\boldsymbol{A}\boldsymbol{W}_1+\Delta\boldsymbol{B}\boldsymbol{N}_1$$

Observe now that the second term in (4.21) can be rewritten as

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}}^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_2^T & \boldsymbol{H}_1^T & \mathbf{0} \\ \boldsymbol{H}_2^T & \boldsymbol{H}_1^T & \mathbf{0} \end{bmatrix}$$
$$+\begin{bmatrix} \mathbf{0} & \boldsymbol{H}_2 & \boldsymbol{H}_2 \\ \mathbf{0} & \boldsymbol{H}_1 & \boldsymbol{H}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 \end{bmatrix} \prec 0$$

Then, the application of the result of Lemma 8 gives

$$\begin{bmatrix} -\boldsymbol{W}_2+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_1^T & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & \boldsymbol{\Omega}_3 & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 \end{bmatrix}$$

$$+\epsilon^{-1}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 \end{bmatrix}\prec 0$$

where

$$\boldsymbol{\Omega}_3 = \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_1^T$$

Finally, an obvious application of the Schur complement formula gives the LMI (4.19) and the proof is complete. ■

It is clear that to compute the controller matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ it is necessary to compute the inverse of the matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ - a task where numerical problems could well arise if this matrix is badly scaled or almost singular.

One of the options is to obtain the solution with the smallest condition number. The problem of minimizing the condition number of a matrix $\boldsymbol{U}(x) \succ 0$, that depends affinely on the variable x, can be formulated as a GEVP (3.14) of the following form

$$\min \gamma \qquad \text{subject to} \begin{cases} \boldsymbol{F}(x) \succ 0, \\ \mu > 0, \\ \mu\boldsymbol{I} \prec \boldsymbol{U}(x) \prec \gamma\mu\boldsymbol{I} \end{cases} \tag{4.22}$$

where $\boldsymbol{F}(x) \succ 0$ is the LMI constraint. However, the computational complexity of this approach is 'quite high' (due to the fact that solving GEVP typically demands more computational effort than solving other LMI problems) and it turns out that reformulating the problem into EVP makes the computation process more efficient. EVP form is obtained in two steps (see (Boyd *et al.*, 1994) for further details).

Step 1: Rewrite the LMI constraint $\boldsymbol{F}(x) \succ 0$ and the positive definite matrix $\boldsymbol{U}(x)$ as

$$\boldsymbol{F}(x) = \boldsymbol{F}_0 + \sum_{i=1}^{m} x_i\boldsymbol{F}_i, \quad \boldsymbol{U}(x) = \boldsymbol{U}_0 + \sum_{i=1}^{m} x_i\boldsymbol{U}_i$$

Step 2: Define the new variables $\nu = 1/\mu$ and $\widetilde{x} = x/\mu$, which yields the following EVP problem:

$$\min \gamma \qquad \text{subject to} \begin{cases} \nu\boldsymbol{F}_0 + \sum_{i=1}^{m} \widetilde{x}_i\boldsymbol{F}_i \succ 0, \\ \boldsymbol{I} \prec \nu\boldsymbol{U}_0 + \sum_{i=1}^{m} \widetilde{x}_i\boldsymbol{U}_i \prec \gamma\boldsymbol{I} \end{cases} \tag{4.23}$$

The above optimization procedure can be easily combined with many LMI conditions.

4.2.2.2. Polytopic model of uncertainty

Consider now the process which the current pass state updating equation matrices range in the given polytope of matrices as it is described by

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 & \boldsymbol{B} \end{bmatrix} \in \operatorname{Co}\left(\begin{bmatrix} \boldsymbol{A}^i & \boldsymbol{B}_0^i & \boldsymbol{B}^i \end{bmatrix}\right),\ i=1,2,\ldots,h \tag{4.24}$$

and

$$\operatorname{Co}\left(\begin{bmatrix} \boldsymbol{A}^i\ \boldsymbol{B}^i\ \boldsymbol{B}_0^i \end{bmatrix}\right) := \left\{ \boldsymbol{X} : \boldsymbol{X} = \sum_{i=1}^{h} \alpha_i \begin{bmatrix} \boldsymbol{A}^i\ \boldsymbol{B}^i\ \boldsymbol{B}_0^i \end{bmatrix},\ \alpha_i \geq 0,\ \sum_{i=1}^{h} \alpha_i = 1 \right\} \tag{4.25}$$

For the current pass profile updating equation in (4.3) we assume norm-bounded type of uncertainty, i.e.

$$y_{k+1}(t) = (\boldsymbol{C}+\Delta\boldsymbol{C})x_{k+1}(t) + (\boldsymbol{D}_0+\Delta\boldsymbol{D}_0)y_k(t) + (\boldsymbol{D}+\Delta\boldsymbol{D})u_{k+1}(t) \tag{4.26}$$

where

$$\begin{bmatrix} \Delta\boldsymbol{C}\ \Delta\boldsymbol{D}_0\ \Delta\boldsymbol{D} \end{bmatrix} = \boldsymbol{H}_2\boldsymbol{\mathcal{F}}\begin{bmatrix} \boldsymbol{E}_1\ \boldsymbol{E}_2\ \boldsymbol{E}_3 \end{bmatrix} \tag{4.27}$$

and the matrix $\boldsymbol{\mathcal{F}}$ satisfies (4.5).

Theorem 4.4. *Suppose that a differential LRP of the form described by (4.3), with uncertainty structure modelled by (4.24)- (4.25) and (4.27), is subjected to a control law of the form of (4.17). Then the resulting closed-loop process is stable along the pass for all admissible uncertainties if there exist matrices $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ of compatible dimensions and a scalar $\epsilon > 0$, such that*

$$\begin{bmatrix} -\boldsymbol{W}_2+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & \boldsymbol{W}_1\boldsymbol{A}^{iT}+\boldsymbol{N}_1^T\boldsymbol{B}^{iT}+\boldsymbol{A}^i\boldsymbol{W}_1+\boldsymbol{B}^i\boldsymbol{N}_1 & \boldsymbol{B}_0^i\boldsymbol{W}_2+\boldsymbol{B}^i\boldsymbol{N}_2 & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \boldsymbol{0} \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^{iT}+\boldsymbol{N}_2^T\boldsymbol{B}^{iT} & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T \\ \boldsymbol{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 & \boldsymbol{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \tag{4.28}$$

If the LMI (4.28) holds then the controller matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are given by (4.20).

Proof. *Taking account of the results of Theorem 4.2 and Theorem 4.3, it is clear that the following inequality holds*

$$\begin{bmatrix} -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & \boldsymbol{W}_1\boldsymbol{A}^{iT}+\boldsymbol{N}_1^T\boldsymbol{B}^{iT}+\boldsymbol{A}^i\boldsymbol{W}_1+\boldsymbol{B}^i\boldsymbol{N}_1 & \boldsymbol{B}_0^i\boldsymbol{W}_2+\boldsymbol{B}^i\boldsymbol{N}_2 \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^{iT}+\boldsymbol{N}_2^T\boldsymbol{B}^{iT} & -\boldsymbol{W}_2 \end{bmatrix}$$

$$+\begin{bmatrix} \boldsymbol{0} & \Delta\boldsymbol{C}\boldsymbol{W}_1+\Delta\boldsymbol{D}\boldsymbol{N}_1 & \Delta\boldsymbol{D}_0\boldsymbol{W}_2+\Delta\boldsymbol{D}\boldsymbol{N}_2 \\ \boldsymbol{W}_1\Delta\boldsymbol{C}^T+\boldsymbol{N}_1^T\Delta\boldsymbol{D}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}_2\Delta\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\Delta\boldsymbol{D}^T & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \prec 0$$

Obviously, the second term in the above inequality is nonlinear in matrix variables $\{\boldsymbol{N}_1,\ \boldsymbol{N}_2,\ \boldsymbol{W}_1,\ \boldsymbol{W}_2\}$ and unknown matrices $\{\Delta\boldsymbol{C}, \Delta\boldsymbol{D}_0,\ \Delta\boldsymbol{D}\}$. Therefore, the robust stabilisation problem considered here can be stated in terms of BMI, for which polynomial-time interior-point algorithms cannot be applied. To overcome this problem, i.e. to find the LMI problem formulation, rewrite the second term as

$$\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\mathcal{F}}^T & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\mathcal{F}}^T & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}}^T\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \boldsymbol{H}_2^T & \mathbf{0} & \mathbf{0}\\ \boldsymbol{H}_2^T & \mathbf{0} & \mathbf{0}\end{bmatrix}$$
$$+\begin{bmatrix}\mathbf{0} & \boldsymbol{H}_2 & \boldsymbol{H}_2\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2\end{bmatrix}\prec 0$$

Now, it is straightforward to see that application of Lemma 8 followed by carrying out the Schur complement gives (4.28). This completes the proof. ■

4.3. Robust stability and stabilisation of discrete LRPs

In this section, LMI methods are used as a basis to provide solutions of robust stability and robust stabilisation problems for discrete LRPs. Based on the state-space model of such processes, robust stability and robust stabilisation (using an appropriately specified control law) conditions are provided for solving them in terms of the feasibility of some LMIs.

4.3.1. Robust stability

The following state-space model of discrete LRPs is considered

$$\begin{aligned} x_{k+1}(p+1) &= (\boldsymbol{A}+\Delta\boldsymbol{A})x_{k+1}(p)+(\boldsymbol{B}_0+\Delta\boldsymbol{B}_0)y_k(p)+(\boldsymbol{B}+\Delta\boldsymbol{B})u_{k+1}(p)\\ y_{k+1}(p) &= (\boldsymbol{C}+\Delta\boldsymbol{C})x_{k+1}(p)+(\boldsymbol{D}_0+\Delta\boldsymbol{D}_0)y_k(p)+(\boldsymbol{D}+\Delta\boldsymbol{D})u_{k+1}(p) \end{aligned} \tag{4.29}$$

The matrices $\Delta\boldsymbol{A}$, $\Delta\boldsymbol{B}$, $\Delta\boldsymbol{B}_0$, $\Delta\boldsymbol{C}$, $\Delta\boldsymbol{D}$, $\Delta\boldsymbol{D}_0$ represent admissible uncertainties to be of the form

$$\begin{bmatrix}\Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0 & \Delta\boldsymbol{B}\\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 & \Delta\boldsymbol{D}\end{bmatrix}=\begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\end{bmatrix}\boldsymbol{\mathcal{F}}\begin{bmatrix}\boldsymbol{E}_1 & \boldsymbol{E}_2 & \boldsymbol{E}_3\end{bmatrix} \tag{4.30}$$

where $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ are some known constant matrices with compatible dimensions and $\boldsymbol{\mathcal{F}}$ is an unknown constant matrix which satisfies (4.5).

Now we have the following sufficient condition for stability along the pass in terms of LMI (the LMI feasibility problem).

Theorem 4.5. *An unforced discrete LRP described by (4.29) is stable along the pass for all admissible uncertainties if there exist matrices $\boldsymbol{P}_1\succ 0$, $\boldsymbol{P}_2\succ 0$ and a* *scalar $\epsilon > 0$ such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{P}_1 & \mathbf{0} & \boldsymbol{P}_1\boldsymbol{A} & \boldsymbol{P}_1\boldsymbol{B}_0 & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \mathbf{0} & -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \boldsymbol{A}^T\boldsymbol{P}_1 & \boldsymbol{C}^T\boldsymbol{P}_2 & -\boldsymbol{P}_1+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{B}_0^T\boldsymbol{P}_1 & \boldsymbol{D}_0^T\boldsymbol{P}_2 & \mathbf{0} & -\boldsymbol{P}_2+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \quad (4.31)$$

Proof. *First, define the vectors*

$$\xi(k+1,p) = \begin{bmatrix} x_{k+1}(p+1) \\ y_{k+1}(p) \end{bmatrix}, \quad \zeta(k,p) = \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \quad (4.32)$$

and use the matrices (4.6) and (4.7) to rewrite the state-space model (4.29) as

$$\xi(k+1,p) = ((\boldsymbol{A}_1 + \Delta\boldsymbol{A}_1) + (\boldsymbol{A}_2 + \Delta\boldsymbol{A}_2))\,\zeta(k,p) \quad (4.33)$$

Further, choose the candidate Lyapunov function as that defined in (2.35). Since

$$\begin{aligned} \Delta V_1(k,p) =& x_{k+1}^T(p+1)\boldsymbol{P}_1 x_{k+1}(p+1) - x_{k+1}^T(p)\boldsymbol{P}_1 x_{k+1}(p) \\ \Delta V_2(k,p) =& y_{k+1}^T(p)\boldsymbol{P}_2 y_{k+1}(p) - y_k^T(p)\boldsymbol{P}_2 y_k(p) \end{aligned}$$

then the increment is

$$\begin{aligned} \Delta V(k,p) =& \Delta V_1(k,p) + \Delta V_2(k,p) \\ =& x_{k+1}^T(p+1)\boldsymbol{P}_1 x_{k+1}(p+1) - x_{k+1}^T(p)\boldsymbol{P}_1 x_{k+1}(p) \\ & + y_{k+1}^T(p)\boldsymbol{P}_2 y_{k+1}(p) - y_k^T(p)\boldsymbol{P}_2 y_k(p) \end{aligned} \quad (4.34)$$

which together with (4.32) and (4.33) gives

$$\begin{aligned} \Delta V(k,p) =& \zeta^T(k,p)\,\big((\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)^T\boldsymbol{P}(\boldsymbol{A}_1+\Delta\boldsymbol{A}_1) \\ & + (\boldsymbol{A}_2 + \Delta\boldsymbol{A}_2)\boldsymbol{P}(\boldsymbol{A}_2 + \Delta\boldsymbol{A}_2) - \boldsymbol{P}\big)\,\zeta(k,p) \end{aligned} \quad (4.35)$$

where $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{P}_1, \boldsymbol{P}_2)$. Hence stability along the pass holds if $\Delta V(k,p) < 0$ for $\forall\zeta(k,p) \neq 0$. Next, use of the Schur complement formula followed by application of Lemma 8 yield

$$\begin{bmatrix} -\boldsymbol{P}_1 & \mathbf{0} & \boldsymbol{P}_1\boldsymbol{A} & \boldsymbol{P}_1\boldsymbol{B}_0 \\ \mathbf{0} & -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{A}^T\boldsymbol{P}_1 & \boldsymbol{C}^T\boldsymbol{P}_2 & -\boldsymbol{P}_1+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \mathbf{0} \\ \boldsymbol{B}_0^T\boldsymbol{P}_1 & \boldsymbol{D}_0^T\boldsymbol{P}_2 & \mathbf{0} & -\boldsymbol{P}_2+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 \end{bmatrix} + \epsilon^{-1}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \mathbf{0} & \mathbf{0} & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix} \prec 0$$

Finally, using the Schur complement formula, we find that the last inequality is equivalent to the LMI (4.31). This completes the proof. ■

The interesting point to note is that the alternative problem formulation provides us with a procedure to increase the process robustness. To proceed, rewrite the uncertain process (4.29) as

$$\begin{aligned}\begin{bmatrix} x_{k+1}(p+1) \\ y_{k+1}(p) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix} \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} + \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} w_{k+1}(p) \\ z_{k+1}(p) &= \begin{bmatrix} \boldsymbol{E}_1 \; \boldsymbol{E}_2 \end{bmatrix} \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \\ w_{k+1}(p) &= \gamma^{-1} \boldsymbol{\mathcal{F}} z_{k+1}(p) \end{aligned} \tag{4.36}$$

which is a feedback system - for illustration see Fig. 4.3. Hence, based on (4.34)

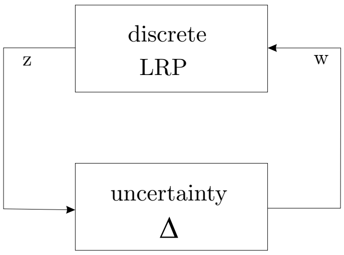


Fig. 4.3. An uncertain process as a feedback system.

and (4.36) we have that

$$\begin{aligned}\Delta V(k,p) =& \left(\zeta^T(k,p)\boldsymbol{A}_1^T + w_{k+1}^T(p)\overline{\boldsymbol{H}}_1^T\right)\boldsymbol{P}\left(\boldsymbol{A}_1\zeta(k,p) + \overline{\boldsymbol{H}}_1 w_{k+1}(p)\right) \\ &+ \left(\zeta^T(k,p)\boldsymbol{A}_2^T + w_{k+1}^T(p)\overline{\boldsymbol{H}}_2^T\right)\boldsymbol{P}\left(\boldsymbol{A}_2\zeta(k,p) + \overline{\boldsymbol{H}}_2 w_{k+1}(p)\right) - \zeta^T(k,p)\boldsymbol{P}\zeta(k,p) \\ =& \begin{bmatrix} \zeta(k,p) \\ w_{k+1}(p) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{A}_1^T\boldsymbol{P}\boldsymbol{A}_1 + \boldsymbol{A}_2^T\boldsymbol{P}\boldsymbol{A}_2 - \boldsymbol{P} & \boldsymbol{A}_1^T\boldsymbol{P}\overline{\boldsymbol{H}}_1 + \boldsymbol{A}_2^T\boldsymbol{P}\overline{\boldsymbol{H}}_2 \\ \overline{\boldsymbol{H}}_1^T\boldsymbol{P}\boldsymbol{A}_1 + \overline{\boldsymbol{H}}_2^T\boldsymbol{P}\boldsymbol{A}_2 & \overline{\boldsymbol{H}}_1^T\boldsymbol{P}\overline{\boldsymbol{H}}_1 + \overline{\boldsymbol{H}}_2^T\boldsymbol{P}\overline{\boldsymbol{H}}_2 \end{bmatrix} \begin{bmatrix} \zeta(k,p) \\ w_{k+1}(p) \end{bmatrix}\end{aligned}$$

where

$$\overline{\boldsymbol{H}}_1 = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{0} \end{bmatrix}, \quad \overline{\boldsymbol{H}}_2 = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{H}_2 \end{bmatrix}$$

Recalling that stability along the pass is held if $\Delta V(k,p) < 0$ for $\forall \{\zeta(k,p), w_{k+1}(p)\} \neq 0$. Therefore $\Delta V(k,p) < 0$ is satisfied when

$$\begin{bmatrix} \boldsymbol{A}_1^T\boldsymbol{P}\boldsymbol{A}_1 + \boldsymbol{A}_2^T\boldsymbol{P}\boldsymbol{A}_2 - \boldsymbol{P} & \boldsymbol{A}_1^T\boldsymbol{P}\overline{\boldsymbol{H}}_1 + \boldsymbol{A}_2^T\boldsymbol{P}\overline{\boldsymbol{H}}_2 \\ \overline{\boldsymbol{H}}_1^T\boldsymbol{P}\boldsymbol{A}_1 + \overline{\boldsymbol{H}}_2^T\boldsymbol{P}\boldsymbol{A}_2 & \overline{\boldsymbol{H}}_1^T\boldsymbol{P}\overline{\boldsymbol{H}}_1 + \overline{\boldsymbol{H}}_2^T\boldsymbol{P}\overline{\boldsymbol{H}}_2 \end{bmatrix} \prec 0 \tag{4.37}$$

Next, observe that (4.5) implies that

$$\gamma^{-2}\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}} \preceq \gamma^{-2}\boldsymbol{I}$$

hence

$$w_{k+1}^T(p)w_{k+1}(p) = \gamma^{-2}z_{k+1}^T(p)\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{F}}z_{k+1}(p) \preceq \gamma^{-2}z_{k+1}^T(p)z_{k+1}(p)$$

Then, it is straightforward to see that for any $w_{k+1}(p)$, $z_{k+1}(p)$ and $\forall\gamma > 0$ the following holds

$$w_{k+1}^T(p)w_{k+1}(p) - \gamma^{-2}z_{k+1}^T(p)z_{k+1}(p) \leq 0$$

thus by noting $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{E}_1 \; \boldsymbol{E}_2 \end{bmatrix}$ we have

$$\begin{aligned} & w_{k+1}^T(p)w_{k+1}(p) - \gamma^{-2}z_{k+1}^T(p)z_{k+1}(p) \\ & = \begin{bmatrix} \zeta^T(k,p) \; w_{k+1}^T(p) \end{bmatrix} \begin{bmatrix} -\boldsymbol{E}^T\boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & \gamma^{-2}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \zeta(k,p) \\ w_{k+1}(p) \end{bmatrix} \leq 0 \end{aligned} \tag{4.38}$$

Combining the results of (4.37) and (4.38) yields the final LMI condition for robust stability of discrete LRPs as

$$\begin{bmatrix} \boldsymbol{A}_1^T\boldsymbol{P}\boldsymbol{A}_1 + \boldsymbol{A}_2^T\boldsymbol{P}\boldsymbol{A}_2 - \boldsymbol{P} + \boldsymbol{E}^T\boldsymbol{E} & \boldsymbol{A}_1^T\boldsymbol{P}\overline{\boldsymbol{H}}_1 + \boldsymbol{A}_2^T\boldsymbol{P}\overline{\boldsymbol{H}}_2 \\ \overline{\boldsymbol{H}}_1^T\boldsymbol{P}\boldsymbol{A}_1 + \overline{\boldsymbol{H}}_2^T\boldsymbol{P}\boldsymbol{A}_2 & \overline{\boldsymbol{H}}_1^T\boldsymbol{P}\overline{\boldsymbol{H}}_1 + \overline{\boldsymbol{H}}_2^T\boldsymbol{P}\overline{\boldsymbol{H}}_2 - \gamma^{-2}\boldsymbol{I} \end{bmatrix} \prec 0 \tag{4.39}$$

It is observed that the term γ in the LMI of (4.39) can be minimized by using linear objective minimization procedure

$$\begin{aligned} & \min_{\boldsymbol{P}\succ 0} \; \mu \\ & \text{subject to (4.39) with } \mu = \gamma^{-2} \end{aligned}$$

which increases the robustness of an uncertain process.

4.3.2. Robust stabilisation

The previous subsection was concerned with robust stability of open-loop processes. Here, a natural extension of obtained results to the case of closed-loop process under a static feedback is presented. The problem here is to find a controller of the form

$$u_{k+1}(p) = \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix} \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \tag{4.40}$$

where $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are appropriately dimensioned matrices to be designed, such that the closed-loop process is robustly stable. It is seen that the control law (4.40) is composed of the weighted sum of current pass state feedback and feedforward of the previous pass profile (see (Gałkowski *et al.*, 2002d) for further background on this form of control action).

Application of the control law (4.40) to (4.29) yields the closed-loop process

$$\begin{aligned} \begin{bmatrix} x_{k+1}(p+1) \\ y_{k+1}(p) \end{bmatrix} = & \left(\begin{bmatrix} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{B}_0+\boldsymbol{B}\boldsymbol{K}_2 \\ \boldsymbol{C}+\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{D}_0+\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix} \right. \\ & \left. + \begin{bmatrix} \Delta\boldsymbol{A}+\Delta\boldsymbol{B}\boldsymbol{K}_1 & \Delta\boldsymbol{B}_0+\Delta\boldsymbol{B}\boldsymbol{K}_2 \\ \Delta\boldsymbol{C}+\Delta\boldsymbol{D}\boldsymbol{K}_1 & \Delta\boldsymbol{D}_0+\Delta\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix} \right) \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \end{aligned} \tag{4.41}$$

where the admissible uncertainties are assumed to be of the form (4.30) and (4.5).

The existence of robustly stabilising $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ can be characterized in LMI terms as follows.

Theorem 4.6. *Suppose that a discrete LRP of the form described by (4.29), with uncertainty structure modelled by (4.30) and (4.5) is subjected to a control law of the form (4.40). Then the resulting closed-loop process is stable along the pass for all admissible uncertainties if there exist matrices* $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$ *and* $\boldsymbol{N}_2$ *of compatible dimensions and a scalar* $\epsilon > 0$ *such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{W}_1 + 2\epsilon \boldsymbol{H}_1\boldsymbol{H}_1^T & 2\epsilon\boldsymbol{H}_2\boldsymbol{H}_1^T & \boldsymbol{A}\boldsymbol{W}_1{+}\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2{+}\boldsymbol{B}\boldsymbol{N}_2 & \mathbf{0} & \mathbf{0} \\ 2\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & -\boldsymbol{W}_2 + 2\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1{+}\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2{+}\boldsymbol{D}\boldsymbol{N}_2 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}_1\boldsymbol{A}^T{+}\boldsymbol{N}_1^T\boldsymbol{B}^T & \boldsymbol{W}_1\boldsymbol{C}^T{+}\boldsymbol{N}_1^T\boldsymbol{D}^T & -\boldsymbol{W}_1 & \mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T{+}\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \mathbf{0} \\ \boldsymbol{W}_2\boldsymbol{B}_0^T{+}\boldsymbol{N}_2^T\boldsymbol{B}^T & \boldsymbol{W}_2\boldsymbol{D}_0^T{+}\boldsymbol{N}_2\boldsymbol{D}^T & \mathbf{0} & -\boldsymbol{W}_2 & \mathbf{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T{+}\boldsymbol{N}_2^T\boldsymbol{E}_3^T \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1{+}\boldsymbol{E}_3\boldsymbol{N}_1 & \mathbf{0} & -\epsilon\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}_2{+}\boldsymbol{E}_3\boldsymbol{N}_2 & \mathbf{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \tag{4.42}$$

and the required controller matrices in (4.40) are given by (4.20).

Proof. *Based on Theorem 4.5 we conclude that the closed-loop process is robustly stabilised by the control law (4.40) if the following matrix inequality is satisfied*

$$\begin{bmatrix} -\boldsymbol{P}_1 & \mathbf{0} & \boldsymbol{P}_1\boldsymbol{A}{+}\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{P}_1\boldsymbol{B}_0{+}\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{K}_2 \\ \mathbf{0} & -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C}{+}\boldsymbol{P}_2\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{P}_2\boldsymbol{D}_0{+}\boldsymbol{P}_2\boldsymbol{D}\boldsymbol{K}_2 \\ \boldsymbol{A}^T\boldsymbol{P}_1{+}\boldsymbol{K}_1^T\boldsymbol{B}^T\boldsymbol{P}_1 & \boldsymbol{C}^T\boldsymbol{P}_2{+}\boldsymbol{K}_1^T\boldsymbol{D}^T\boldsymbol{P}_2 & -\boldsymbol{P}_1 & \mathbf{0} \\ \boldsymbol{B}_0^T\boldsymbol{P}_1{+}\boldsymbol{K}_2^T\boldsymbol{B}^T\boldsymbol{P}_1 & \boldsymbol{D}_0^T\boldsymbol{P}_2{+}\boldsymbol{K}_2\boldsymbol{D}^T\boldsymbol{P}_1 & \mathbf{0} & -\boldsymbol{P}_2 \end{bmatrix}$$
$$+ \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_1^T + \boldsymbol{K}_1^T\boldsymbol{E}_3^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2^T + \boldsymbol{K}_2^T\boldsymbol{E}_3^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}}^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$
$$+ \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \mathbf{0} & \mathbf{0} & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_1 + \boldsymbol{E}_3\boldsymbol{K}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2 + \boldsymbol{E}_3\boldsymbol{K}_2 \end{bmatrix} \prec 0$$

The above inequality is clearly stated in the form of BMI. To reformulate it into LMIs, make use of the following change of variables $\boldsymbol{W}_1 = \boldsymbol{P}_1^{-1}$, $\boldsymbol{W}_2 = \boldsymbol{P}_2^{-1}$ *and then pre- and post- multiply both sides of this last inequality by* $\mathrm{diag}(\boldsymbol{W}_1, \boldsymbol{W}_2,$

$\boldsymbol{W}_1, \boldsymbol{W}_2$). Next, an application of the result of Lemma 8 leads to

$$\epsilon^{-1}\begin{bmatrix} \mathbf{0}\,\mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}\,\mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}\,\mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \mathbf{0} \\ \mathbf{0}\,\mathbf{0} & \mathbf{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T \end{bmatrix}\begin{bmatrix} \mathbf{0}\,\mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}\,\mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}\,\mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \mathbf{0} \\ \mathbf{0}\,\mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 \end{bmatrix}$$
$$+\begin{bmatrix} -\boldsymbol{W}_1+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_1^T & 2\epsilon\boldsymbol{H}_2\boldsymbol{H}_1^T & \boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 \\ 2\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & -\boldsymbol{W}_2+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 \\ \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T & \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & -\boldsymbol{W}_1 & \mathbf{0} \\ \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2\boldsymbol{D}^T & \mathbf{0} & -\boldsymbol{W}_2 \end{bmatrix}\prec 0$$

where $\boldsymbol{N}_1 = \boldsymbol{K}_1\boldsymbol{W}_1$ and $\boldsymbol{N}_2 = \boldsymbol{K}_2\boldsymbol{W}_2$. Finally, the Schur complement formula gives (4.42) and the proof is complete. ■

4.3.2.1. Alternative robust stabilisation

Here another approach to characterize and solve the robust stabilisation problem of discrete LRPs is presented. To proceed, rewrite (4.41) in the form

$$\begin{bmatrix} x_{k+1}(p+1) \\ y_{k+1}(p) \end{bmatrix} = \overline{\boldsymbol{A}}\begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \tag{4.43}$$

where

$$\overline{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{B}_0+\boldsymbol{B}\boldsymbol{K}_2 \\ \boldsymbol{C}+\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{D}_0+\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix} + \begin{bmatrix} \Delta\boldsymbol{A}+\Delta\boldsymbol{B}\boldsymbol{K}_1 & \Delta\boldsymbol{B}_0+\Delta\boldsymbol{B}\boldsymbol{K}_2 \\ \Delta\boldsymbol{C}+\Delta\boldsymbol{D}\boldsymbol{K}_1 & \Delta\boldsymbol{D}_0+\Delta\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix}$$

Suppose also that the matrices describing the uncertainty in this last model are written in the form

$$\begin{aligned} \begin{bmatrix} \Delta\boldsymbol{A}+\Delta\boldsymbol{B}\boldsymbol{K}_1 & \Delta\boldsymbol{B}_0+\Delta\boldsymbol{B}\boldsymbol{K}_2 \\ \Delta\boldsymbol{C}+\Delta\boldsymbol{D}\boldsymbol{K}_1 & \Delta\boldsymbol{D}_0+\Delta\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix}\gamma^{-1}\boldsymbol{\mathcal{F}}\begin{bmatrix} \boldsymbol{E}_1+\boldsymbol{E}_3\boldsymbol{K}_1 & \boldsymbol{E}_2+\boldsymbol{E}_3\boldsymbol{K}_2 \end{bmatrix} \\ &= \gamma^{-1}\boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E} \end{aligned} \tag{4.44}$$

where $\boldsymbol{\mathcal{F}}$ satisfies (4.5). The design parameter γ here can be considered as a term which is used to attenuate the effects of the uncertainty, for which we have the following result.

Theorem 4.7. *Suppose that a discrete LRP of the form described by (4.29), with uncertainty structure modelled by (4.44) and (4.5), is subjected to a control law of the form (4.40). Then the resulting closed-loop process is stable along the pass for all admissible uncertainties if there exist matrices $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$*

of compatible dimensions and a scalar $\gamma > 0$ such that the following LMI holds

$$\begin{gathered}\left[\begin{array}{ccc} -\boldsymbol{W}_1 & \mathbf{0} & \boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1 \\ \mathbf{0} & -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 \\ \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T & \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & -\boldsymbol{W}_1 \\ \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \mathbf{0} \\ \boldsymbol{H}_1^T & \boldsymbol{H}_2^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 \end{array}\right. \\ \left.\begin{array}{ccc} \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{H}_1 & \mathbf{0} \\ \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{H}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T \\ -\boldsymbol{W}_2 & \mathbf{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T \\ \mathbf{0} & -\gamma^2\boldsymbol{I} & \mathbf{0} \\ \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 & \mathbf{0} & -\boldsymbol{I} \end{array}\right] \prec 0\end{gathered} \tag{4.45}$$

Proof. *Using the RM and stability condition for a 2-D system, it follows immediately that the LRP of the form (4.43) is stable along the pass if there exists a block-diagonal matrix $\boldsymbol{P} = diag(\boldsymbol{P}_1, \boldsymbol{P}_2) \succ 0$ such that the following LMI holds*

$$\overline{\boldsymbol{A}}^T\boldsymbol{P}\overline{\boldsymbol{A}} - \boldsymbol{P} \prec 0$$

An obvious application of the Schur complement formula yields

$$\begin{bmatrix} -\boldsymbol{P}^{-1} & \boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E} \\ \boldsymbol{\Omega}^T + \gamma^{-1}\boldsymbol{E}^T\boldsymbol{\mathcal{F}}^T\boldsymbol{H}^T & -\boldsymbol{P} \end{bmatrix} \prec 0$$

where

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{B}_0+\boldsymbol{B}\boldsymbol{K}_2 \\ \boldsymbol{C}+\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{D}_0+\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix}, \ \boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix}, \ \boldsymbol{E} = \begin{bmatrix} \boldsymbol{E}_1+\boldsymbol{E}_3\boldsymbol{K}_1 & \boldsymbol{E}_2+\boldsymbol{E}_3\boldsymbol{K}_2 \end{bmatrix}$$

Applying the result of Lemma 8 to this last condition and then pre- and post-multiplying the result by $\mathrm{diag}(\epsilon^{-\frac{1}{2}}\boldsymbol{P}, \epsilon^{-\frac{1}{2}}\boldsymbol{I})$ *gives*

$$\begin{bmatrix} -\overline{\boldsymbol{P}} + \overline{\boldsymbol{P}}\gamma^{-2}\boldsymbol{H}\boldsymbol{H}^T\overline{\boldsymbol{P}} & \overline{\boldsymbol{P}}\boldsymbol{\Omega} \\ \boldsymbol{\Omega}^T\overline{\boldsymbol{P}} & -\overline{\boldsymbol{P}} + \boldsymbol{E}^T\boldsymbol{E} \end{bmatrix} \prec 0$$

where a new matrix variable $\overline{\boldsymbol{P}}$ is introduced as $\overline{\boldsymbol{P}} = \epsilon^{-1}\boldsymbol{P}$. Next, an application of the Schur complement formula gives

$$\begin{bmatrix} -\overline{\boldsymbol{P}} & \overline{\boldsymbol{P}}\boldsymbol{\Omega} & \overline{\boldsymbol{P}}\boldsymbol{H} & \mathbf{0} \\ \boldsymbol{\Omega}^T\overline{\boldsymbol{P}} & -\overline{\boldsymbol{P}} & \mathbf{0} & \boldsymbol{E}^T \\ \boldsymbol{H}^T\overline{\boldsymbol{P}} & \mathbf{0} & -\gamma^2\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{E} & \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \prec 0 \tag{4.46}$$

Finally, the proof can be completed in an identical manner to the proof of Theorem 4.6. ■

The term γ in the LMI of (4.45) can again be minimized by using a linear objective minimization procedure

$$\begin{gathered}\min_{\boldsymbol{W}_1\succ 0,\boldsymbol{W}_2\succ 0,\boldsymbol{N}_1,\boldsymbol{N}_2} \quad \mu \\ \text{subject to (4.45) with } \mu=\gamma^2\end{gathered}$$

which, due to the presence of the term γ^{-1} in the uncertainty model (4.44), provides an essential advantage as it allows the extension of the uncertainty borders, i.e. it increases the robustness.

4.4. Application examples

4.4.1. Analysis of ILC processes

It is clear that the principal requirement for ILC process is its stability along the direction of learning iterations in addition to the stability of a control system. To proceed, make use of the following change of variables

$$\begin{aligned} r =& k-1 \\ \nu_r(t) =& \nu_{k-1}(t) = e_k(t)\end{aligned}$$

to rewrite (2.53) in the differential LRP form as

$$\begin{bmatrix} \dot{\eta}_{r+1}(t) \\ \nu_{r+1}(t)\end{bmatrix} = \begin{bmatrix} \boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_1 & -\boldsymbol{B}\boldsymbol{K}_2 \\ -\boldsymbol{C}\boldsymbol{A}+\boldsymbol{C}\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{I}-\boldsymbol{C}\boldsymbol{B}\boldsymbol{K}_2\end{bmatrix}\begin{bmatrix}\eta_r(t) \\ \nu_r(t)\end{bmatrix} \tag{4.47}$$

Since stability of the model (4.47) guarantees the learning convergence, then the result of Theorem 4.3 provides the LMI condition for stability checking. Hence, omitting the uncertainty, the LMI condition for stability ILC process can be formulated as follows

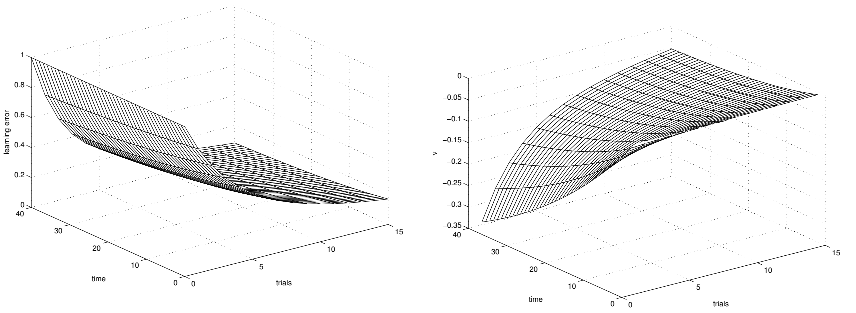
$$\begin{bmatrix} -\boldsymbol{W}_2 & -\boldsymbol{C}\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{C}\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{W}_2-\boldsymbol{C}\boldsymbol{B}\boldsymbol{N}_2 \\ -\boldsymbol{W}_1\boldsymbol{A}^T\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T\boldsymbol{C}^T & \boldsymbol{W}_1\boldsymbol{A}^T-\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{A}\boldsymbol{W}_1-\boldsymbol{B}\boldsymbol{N}_1 & -\boldsymbol{B}\boldsymbol{N}_2 \\ \boldsymbol{W}_2-\boldsymbol{N}_2^T\boldsymbol{B}^T\boldsymbol{C}^T & -\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2\end{bmatrix}\prec 0 \tag{4.48}$$

where $\boldsymbol{W}_1\succ 0$, $\boldsymbol{W}_2\succ 0$, $\boldsymbol{N}_1$, $\boldsymbol{N}_2$ are the matrices to be found. If they exist then the explicit specification for the learning gain matrices are given by (4.20). It is significant to note that the LMI condition of (4.48) can be easily implemented and therefore tools like Lmi Control Toolbox, SeDuMi, or any equivalent software can be used to solve the addressed problem.

To show the usefulness of learning gain matrices design method, the following numerical example is given.

Example 4.3. *Let us consider the ILC process (2.49) with the matrices given by*

$$\boldsymbol{A}=\begin{bmatrix}0.1341 & 0.2674 \\ 1.1977 & 1.6292\end{bmatrix},\ \boldsymbol{B}=\begin{bmatrix}0.5643 & 0.6876 \\ 1.8219 & -0.8877\end{bmatrix},\ \boldsymbol{C}=\begin{bmatrix}0.3095 & -0.3533 \\ 0.0438 & 0.3968\end{bmatrix}$$



learning error state (η)

Fig. 4.4. Simulation results.

Then the LMI (4.48) is feasible with

$$\boldsymbol{W}_1=\begin{bmatrix}1.0198 & 0.0000\\ 0.0000 & 1.0198\end{bmatrix},\ \boldsymbol{W}_2=\begin{bmatrix}0.6579 & -0.0251\\ -0.0251 & 0.6458\end{bmatrix},$$

$$\boldsymbol{N}_1=\begin{bmatrix}0.8024 & 0.9749\\ 0.2487 & -0.3750\end{bmatrix},\ \boldsymbol{N}_2=\begin{bmatrix}0.0165 & 0.1004\\ 0.2332 & -0.0297\end{bmatrix}$$

and the learning gain matrices are

$$\boldsymbol{K}_1=\begin{bmatrix}0.7868 & 0.9560\\ 0.2439 & -0.3677\end{bmatrix},\ \boldsymbol{K}_2=\begin{bmatrix}0.0311 & 0.1567\\ 0.3532 & -0.0323\end{bmatrix}$$

To confirm the learning convergence in this case, the simulation results are provided - see Fig. 4.4. It is assumed that for the first trial the learning error is 1.

Furthermore, consider the case when (4.47) is subjected to norm-bounded uncertainty i.e. it is assumed that uncertainty is modelled as an additive perturbation (denoted here by $\Delta\boldsymbol{A}$ and $\Delta\boldsymbol{B}$) to the nominal ILC process matrices ($\boldsymbol{A}$ and $\boldsymbol{B}$)

$$[\Delta\boldsymbol{A}\ \Delta\boldsymbol{B}]=\boldsymbol{H}_1\boldsymbol{\mathcal{F}}\,[\boldsymbol{E}_1\ \boldsymbol{E}_2]$$

where $\boldsymbol{H}_1$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$ are given matrices and the matrix $\boldsymbol{\mathcal{F}}$ satisfies (4.5). Hence the following matrix inequality can provide robust stability condition for the ILC process

$$\begin{bmatrix} -\boldsymbol{W}_2 & -\boldsymbol{C}\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{C}\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{W}_2-\boldsymbol{C}\boldsymbol{B}\boldsymbol{N}_2\\ -\boldsymbol{W}_1\boldsymbol{A}^T\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T\boldsymbol{C}^T & \boldsymbol{W}_1\boldsymbol{A}^T-\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{A}\boldsymbol{W}_1-\boldsymbol{B}\boldsymbol{N}_1 & -\boldsymbol{B}\boldsymbol{N}_2\\ \boldsymbol{W}_2-\boldsymbol{N}_2^T\boldsymbol{B}^T\boldsymbol{C}^T & -\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 \end{bmatrix}$$

$$+\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{W}_1\Delta\boldsymbol{A}^T-\boldsymbol{N}_1^T\Delta\boldsymbol{B}^T+\Delta\boldsymbol{A}\boldsymbol{W}_1-\Delta\boldsymbol{B}\boldsymbol{N}_1 & -\Delta\boldsymbol{B}\boldsymbol{N}_2\\ \boldsymbol{0} & -\boldsymbol{N}_2^T\Delta\boldsymbol{B}^T & \boldsymbol{0} \end{bmatrix}$$

$$+\begin{bmatrix} \boldsymbol{0} & -\boldsymbol{C}\Delta\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{C}\Delta\boldsymbol{B}\boldsymbol{N}_1 & -\boldsymbol{C}\Delta\boldsymbol{B}\boldsymbol{N}_2\\ -\boldsymbol{W}_1\Delta\boldsymbol{A}^T\boldsymbol{C}^T+\boldsymbol{N}_1^T\Delta\boldsymbol{B}^T\boldsymbol{C}^T & \boldsymbol{0} & \boldsymbol{0}\\ -\boldsymbol{N}_2^T\Delta\boldsymbol{B}^T\boldsymbol{C}^T & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\prec 0$$

Unfortunately, the above inequality is nonlinear and cannot be efficiently solved. To make the inequality amenable to polynomial-time computation, the matrix manipulation provided by Theorem 4.3 are applied. As a result, the following LMI condition is obtained

$$\left[\begin{array}{ccccccc}
-\boldsymbol{W}_2 + 2\epsilon_2 \boldsymbol{C}\boldsymbol{H}_1\boldsymbol{H}_1^T\boldsymbol{C}^T & -\boldsymbol{C}\boldsymbol{A}\boldsymbol{W}_1 + \boldsymbol{C}\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{W}_2 - \boldsymbol{C}\boldsymbol{B}\boldsymbol{N}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
-\boldsymbol{W}_1\boldsymbol{A}^T\boldsymbol{C}^T + \boldsymbol{N}_1^T\boldsymbol{B}^T\boldsymbol{C}^T & \boldsymbol{W}_1\boldsymbol{A}^T - \boldsymbol{N}_1^T\boldsymbol{B}^T + \boldsymbol{A}\boldsymbol{W}_1 - \boldsymbol{B}\boldsymbol{N}_1 + 2\epsilon_1\boldsymbol{H}_1\boldsymbol{H}_1^T & -\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{W}_1\boldsymbol{E}_1^T + \boldsymbol{N}_2^T\boldsymbol{E}_2^T & \mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T + \boldsymbol{N}_1^T\boldsymbol{E}_2^T & \mathbf{0} \\
\boldsymbol{W}_2 - \boldsymbol{N}_2^T\boldsymbol{B}^T\boldsymbol{C}^T & -\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 & \mathbf{0} & \boldsymbol{N}_2^T\boldsymbol{E}_2^T & \mathbf{0} & \boldsymbol{N}_2^T\boldsymbol{E}_2^T \\
\mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1 + \boldsymbol{E}_2\boldsymbol{N}_2 & \mathbf{0} & -\epsilon_1\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{N}_2 & \mathbf{0} & -\epsilon_1\boldsymbol{I} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1 + \boldsymbol{E}_2\boldsymbol{N}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\epsilon_2\boldsymbol{I} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{N}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\epsilon_2\boldsymbol{I}
\end{array}\right] \prec 0$$

where the matrices $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$, $\boldsymbol{N}_2$ and the scalars $\epsilon_1 > 0$, $\epsilon_2 > 0$ are to be found via numerical computations.

The following example shows the usefulness of provided LMI condition.

Example 4.4. *Let us consider the same ILC process as in Example 4.3. Further, suppose that the uncertainty is modelled with*

$$\boldsymbol{H}_1 = \begin{bmatrix} 0.0001 \\ 0.0061 \end{bmatrix}, \ \boldsymbol{E}_1 = \begin{bmatrix} 0.0763 & 0.0541 \end{bmatrix}, \ \boldsymbol{E}_2 = \begin{bmatrix} 0.0537 & 0.0331 \end{bmatrix}$$

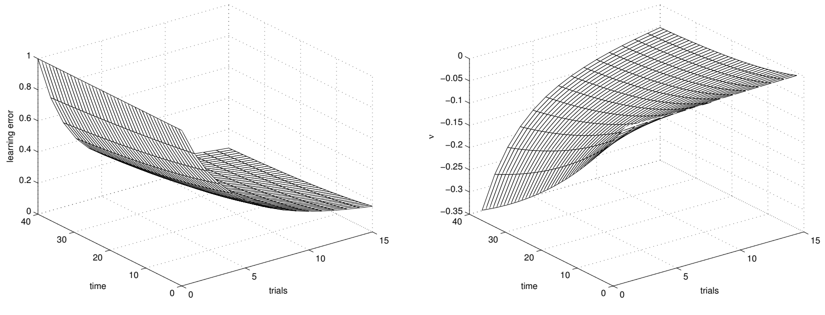
then the following numerical result is obtained

$$\boldsymbol{W}_1 = \begin{bmatrix} 1.0720 & -0.0094 \\ -0.0094 & 1.0796 \end{bmatrix}, \ \boldsymbol{W}_2 = \begin{bmatrix} 0.7009 & -0.0265 \\ -0.0265 & 0.6878 \end{bmatrix},$$
$$\boldsymbol{N}_1 = \begin{bmatrix} 0.8395 & 1.0294 \\ 0.2718 & -0.3933 \end{bmatrix}, \ \boldsymbol{N}_2 = \begin{bmatrix} 0.0194 & 0.1084 \\ 0.2514 & -0.0292 \end{bmatrix}$$

with $\epsilon_1 = 1.0860$ *and* $\epsilon_2 = 1.0860$. *In this case, the corresponding learning gain matrices are*

$$\boldsymbol{K}_1 = \begin{bmatrix} 0.7915 & 0.9603 \\ 0.2504 & -0.3621 \end{bmatrix}, \ \boldsymbol{K}_2 = \begin{bmatrix} 0.0338 & 0.1590 \\ 0.3576 & -0.0286 \end{bmatrix}$$

To confirm that the obtained controller works properly for all models encountered by an assumed uncertainty set, the simple simulation for the particular choice of the matrix $\boldsymbol{\mathcal{F}} = 0.7$ *is presented. Simulations of a learning error* $\nu_{r+1}(t)$ *and a state* $\eta_{r+1}(t)$ *variables are depicted in Fig. 4.5 (with the assumption that the initial learning error is 1). Since the number of decision variables to be found in this example is 16, then time required to find the solution was 0.4 seconds.*



learning error (ν) state (η)

Fig. 4.5 . Simulation results for an uncertain ILC process.

Table 4.1. Computation time.

n	m	p	q	CPU time in seconds
2	2	2	16	0.40
3	2	3	26	0.72
5	2	5	52	1.86
6	5	6	104	7.53
10	7	10	252	71.73
12	10	12	398	259.09

Obviously, due to the polynomial time complexity of the considered problem, efficiency is maintained for higher problem dimensions - see Table 4.1. Note that all computations have been performed with Lmi Control Toolbox *1.0.8 under* Matlab *6.5. The* Matlab-*files have been run on a PC with AMD Duron 600 MHz CPU and 128MB RAM.*

The overall number of decision variables q is computed from

$$q = \frac{n(n+1)}{2} + \frac{m(m+1)}{2} + nl + ml + 2$$

where n is the number of states, m is the number of outputs and l is the number of inputs. Further, the terms sequence in the above equation denotes the number of decision variables involved in $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{N}_1$, $\boldsymbol{N}_2$ and $\epsilon_{1,2}$ respectively.

4.4.2. Stability of a parallel computing process

To consider the stability problem of a parallel computing process, note first that one of the variables n_1 and n_2 can have a finite value (in this case 2) then system (2.55) can be considered and analysed as a discrete LRP.

It is written in (Bauer *et al.*, 2001) that the use of 2-D state-space models for parallel computing processes has been motivated by existing results on the stability of such models. Unfortunately, considered in the paper stability conditions are computationally ineffective due to large dimensions of used matrices. Indeed, in that paper the following matrix is used

$$\boldsymbol{A}_{cc}(n) = \begin{bmatrix} \boldsymbol{J}(n,0) & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{K}(n,0) & \boldsymbol{J}(n-1,1) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{J}(0,n) \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{K}(0,n) \end{bmatrix}$$

where n denotes the time instant and matrices $\boldsymbol{J}(n_1,n_2)$ and $\boldsymbol{K}(n_1,n_2)$ are identifying in (2.55) as

$$\boldsymbol{J}(n_1,n_2) = \begin{bmatrix} \boldsymbol{A}_{11}(n_1,n_2) & \boldsymbol{A}_{12}(n_1,n_2) \\ 0 & 0 \end{bmatrix}, \boldsymbol{K}(n_1,n_2) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_{21}(n_1,n_2) & \boldsymbol{A}_{22}(n_1,n_2) \end{bmatrix}$$

Hence, it is clear that the dimension of the matrix $\boldsymbol{A}_{cc}(n)$ can be vary large due to n value, therefore potential computational and numerical problems may occur. Moreover, it is a necessity to perform computations for all possible choices of incomplete output masks. Therefore, as it is written in (Bauer *et al.*, 2001), the stability problem belongs to the class of $\mathcal{NP}$-hard problems - even for finite n. To overcome these difficulties the approach developed in this chapter can be used to find an exact or an approximate solution to the stability problem of parallel computing processes.

While the number of all possible choices of incomplete output masks is not large, then the stability test can be based on LMI condition provided by Theorem 4.5 where there are no uncertain matrices in the process model (4.3).

On the other hand, it can be assumed that matrices, which are the result of all possible choices of incomplete output masks, belong to some bounded set of the space of matrices. Furthermore, suppose that this set is a convex set or it can be approximated by a convex set represented by $\boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}$ where $\boldsymbol{\mathcal{F}}$ satisfies (4.5) and the matrices $\boldsymbol{H}$ and $\boldsymbol{E}$ are given. That is, matrices uncertainty can be put in the norm-bounded form and hence (2.55) can be written as

$$\begin{bmatrix} x^h(n_1+1,n_2) \\ x^v(n_1,n_2+1) \end{bmatrix} = \left(\begin{bmatrix} \boldsymbol{A}_{11}(n_1,n_2) & \boldsymbol{A}_{12}(n_1,n_2) \\ \boldsymbol{A}_{21}(n_1,n_2) & \boldsymbol{A}_{22}(n_1,n_2) \end{bmatrix} \right. \\ \left. + \begin{bmatrix} \Delta\boldsymbol{A}_{11}(n_1,n_2) & \Delta\boldsymbol{A}_{12}(n_1,n_2) \\ \Delta\boldsymbol{A}_{21}(n_1,n_2) & \Delta\boldsymbol{A}_{22}(n_1,n_2) \end{bmatrix} \right) \begin{bmatrix} x^h(n_1,n_2) \\ x^v(n_1,n_2) \end{bmatrix} \tag{4.49}$$

where

$$\begin{bmatrix} \Delta\boldsymbol{A}_{11}(n_1,n_2) & \Delta\boldsymbol{A}_{12}(n_1,n_2) \\ \Delta\boldsymbol{A}_{21}(n_1,n_2) & \Delta\boldsymbol{A}_{22}(n_1,n_2) \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} \boldsymbol{\mathcal{F}} \begin{bmatrix} \boldsymbol{E}_1 & \boldsymbol{E}_2 \end{bmatrix} = \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}$$

This means that the provided theorems offer a powerful implementable tool to stability analysis of a parallel or distributed computation process. Even though such an approach is applied for each time instant (assumed to be finite), determining the process stability involves quite a reasonable amount of computations.

4.5. Concluding remarks

This chapter shows that LMI methods, which are standard in 1-D system theory, can be applied to analysis and synthesis for LRPs in the presence of uncertainty. The main obstacles towards applying classical 2-D system theory methods in these cases are computational problems which are avoided using LMI methods. Moreover, as it is shown, LMI methods allow us to extend the uncertainty borders via an optimization procedure, i.e. increasing the robustness. Also, they are numerically efficient and allow the efficient handling of considerably high dimensional problems.

Chapter 5

LMI METHODS IN PERFORMANCE ANALYSIS

It has been shown that the condition for the existence of $\mathcal{H}_\infty$ and $\mathcal{H}_2$ controllers can be written in the form of matrix inequality, which turns out to be bilinear with respect to its parameters (Helton and Merino, 1998; Saberi *et al.*, 1995; Zhou *et al.*, 1996). Therefore, such a condition cannot be tested with efficient computational method. Fortunately, we do not have to solve nonlinear problems because potential difficulties can be omitted using a variety of methods. Both Riccati-based (Doyle *et al.*, 1989) and LMI-based (Boyd *et al.*, 1994; Dullerud and Paganini, 2000; Scherer and Weiland, 2002) solutions have been considered for $\mathcal{H}_\infty$ and $\mathcal{H}_2$ control problems in 1-D system theory. It is natural question to ask if such computationally effective approaches can be extended to 2-D (n-D) linear systems and LRPs. In the case of 2-D discrete linear systems, some work on $\mathcal{H}_\infty$ and $\mathcal{H}_2$ approaches have been reported - see, for example, (Du and Xie, 2002; Ŝebek M., 1993; Tuan *et al.*, 2002). However, these control problems for LRPs still remain unsolved although some preliminary results have already been reported on $\mathcal{H}_\infty$ control (Paszke *et al.*, 2003, 2004). This lack of results is mainly seen for differential LRPs, and hence the purpose of this chapter is to fill that gap.

In addition to these results, the solution to the $\mathcal{H}_\infty$ control problem with parameter uncertainty in all the matrices of the process state-space model, is provided. Further, the numerical design procedure for a controller maintaining an adequate level of performance represented by the quadratic cost, is presented. Searching for such a controller is called a guaranteed cost control problem and it has even been considered for 2-D linear systems (Guan *et al.*, 2001). However, this paper only contains a design method for a 2-D system represented by FMM, in which the special case of boundary conditions is taken into account, and it seems that there are no results for LRPs. Therefore, this chapter proposes a design algorithm involving a convex optimization for an LRP controller achieving a suboptimal guaranteed cost such that the process can be stabilised for all admissible uncertainties.

5.1. Performance specifications for LRPs and n-D systems

One of the most frequently considered ways to describe the performance specifications of a control system is to use $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms (Zhou *et al.*, 1996) of signals and systems. These norms turn out to be natural measures of the worst possible performance for many classes of input signals.

In the presented approach, the generalized signals w and z are chosen to characterize performance properties to be achieved by the controller. The signal w is the generalized disturbance which collect various disturbance signals. This signal is assumed to be member of L_2^r space i.e. space of signals with finite energy where the L_2^r norm is defined. In the case of a 2-D signal described by mixed time and space variables, L_2^r norm is defined as follows (see (Du and Xie, 2002) for definition L_2 norm of discrete 2-D signal).

Definition 5.1. *The L_2 norm of the vector $w_k(t) \in \mathbb{R}^{r\times 1}$ defined over $[0,\infty], [0,\infty]$ is given by*

$$\|w\|_2 = \sqrt{\sum_{k=0}^{\infty} \int_0^{\infty} w_k(t)^T w_k(t)\; dt} \tag{5.1}$$

and $w_k(t)$ is said to be a member of $L_2^r\{[0,\infty],[0,\infty]\}$, or L_2^r for short, if $||w||_2 < \infty$.

The signal z represents the controlled variable. Taking into consideration these additional signals, the state-space models of both differential (G_{diff}) and discrete LRPs (G_{disc}) are

$$G_{\text{diff}} \left\{ \begin{array}{rl} \dot{x}_{k+1}(t) = & \boldsymbol{A}x_{k+1}(t) + \boldsymbol{B}_0 y_k(t) + \boldsymbol{B}u_{k+1}(t) + \boldsymbol{B}_1 w_{k+1}(t) \\ y_{k+1}(t) = & \boldsymbol{C}x_{k+1}(t) + \boldsymbol{D}_0 y_k(t) + \boldsymbol{D}u_{k+1}(t) + \boldsymbol{D}_1 w_{k+1}(t) \end{array} \right. \tag{5.2}$$

and

$$G_{\text{disc}} \left\{ \begin{array}{rl} x_{k+1}(p+1) = & \boldsymbol{A}x_{k+1}(p) + \boldsymbol{B}_0 y_k(p) + \boldsymbol{B}u_{k+1}(p) + \boldsymbol{B}_1 w_{k+1}(p) \\ y_{k+1}(p) = & \boldsymbol{C}x_{k+1}(p) + \boldsymbol{D}_0 y_k(p) + \boldsymbol{D}u_{k+1}(p) + \boldsymbol{D}_1 w_{k+1}(p) \end{array} \right. \tag{5.3}$$

respectively. In the above equations, $w_{k+1}(t)$ and $w_{k+1}(p)$ are the $\mathbb{R}^{r\times 1}$ disturbance input vectors which belong to L_2^r. Due to the fact that the pass profile vector is simultaneously the output vector, we can write $z_{k+1}(t) = y_{k+1}(t)$ in differential case or $z_{k+1}(p) = y_{k+1}(p)$ in discrete one (see also the equation (2.21)).

To study $\mathcal{H}_\infty$ or $\mathcal{H}_2$ control of LRPs, definitions of $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms are required. First the $\mathcal{H}_\infty$ norm definition is provided

Definition 5.2. *A differential (discrete) LRP described by (5.2) (by (5.3) respectively) is said to have the $\mathcal{H}_\infty$ disturbance attenuation (the $\mathcal{H}_\infty$ norm bound) γ if it is stable along the pass and*

$$\sup_{0 \neq w \in L_2^r} \frac{\|y\|_2}{\|w\|_2} < \gamma \tag{5.4}$$

The above definition corresponds to the maximum gain peak of frequency response, i.e. supremum of the maximum singular value of the frequency response of the process. This can be computed from

$$\|G(s,z)\|_\infty = \sup_{\omega_1 \in \mathbb{R}, \omega_2 \in [0,2\pi]} \overline{\sigma}\left[G(j\omega_1, e^{j\omega_2})\right]$$

in a differential case and from

$$\|G(z_1,z_2)\|_\infty = \sup_{\omega_1,\omega_2\in[0,2\pi]} \overline{\sigma}\left[G(e^{j\omega_1},e^{j\omega_2})\right]$$

in a discrete case. $G(s,z)$ and $G(z_1,z_2)$ are the 2-D transfer function matrices between pass profile and disturbance signal and they are given by

$$G(s,z) = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} s\boldsymbol{I}-\boldsymbol{A} & -\boldsymbol{B}_0 \\ -z\boldsymbol{C} & \boldsymbol{I}-z\boldsymbol{D}_0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{D}_1 \end{bmatrix} \tag{5.5}$$

and

$$G(z_1,z_2) = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}-z_1\boldsymbol{A} & -z_1\boldsymbol{B}_0 \\ -z_2\boldsymbol{C} & \boldsymbol{I}-z_2\boldsymbol{D}_0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{D}_1 \end{bmatrix} \tag{5.6}$$

for differential and discrete cases respectively. The $\mathcal{H}_2$ norm defined as the square of the L_2 norm of $w_k(t)$ is commonly termed the total energy in the signal $w_k(t)$

$$\|G(s,z)\|_2 = \sqrt{\frac{1}{(2\pi)^2}\int_0^{2\pi}\int_{-\infty}^{\infty} \operatorname{trace}\left(G^*(-j\omega_2,e^{j\omega_1})G(-j\omega_2,e^{j\omega_1})\right)d\omega_2 d\omega_1} \tag{5.7}$$

where $G^*(\cdot)$ denotes the complex conjugate transpose of $G(\cdot)$.

A fundamental point to note is that the $\mathcal{H}_2$ norm of a process coincides with the total output energy in the impulse response of a process. This observation leads immediately to the algorithms that determine the $\mathcal{H}_2$ norm of the process - see Section 5.6 for details.

5.2. $\mathcal{H}_\infty$ control of differential LRPs

5.2.1. LMI-based $\mathcal{H}_\infty$ norm computation

To compute the $\mathcal{H}_\infty$ norm, consider the case of a differential LRP (5.2) with no control inputs but with external disturbance inputs given by

$$\begin{bmatrix} \dot{x}_{k+1}(t) \\ y_{k+1}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix} \begin{bmatrix} x_{k+1}(t) \\ y_k(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{D}_1 \end{bmatrix} w_{k+1}(t) \tag{5.8}$$

Noting that the measured output vector is equal to the pass profile vector, we now have the following Theorem which gives an $\mathcal{H}_\infty$ condition for stability along the pass in terms of LMI.

Theorem 5.1. *A differential LRP which can be written in the form (5.8) is stable along the pass and has the $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$ if there exist matrices $\boldsymbol{P}_1 \succ 0$, and $\boldsymbol{P}_2 \succ 0$ of appropriate dimensions such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 & \boldsymbol{P}_2\boldsymbol{D}_1 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A} & \boldsymbol{P}_1\boldsymbol{B}_0 & \boldsymbol{P}_1\boldsymbol{B}_1 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & -\boldsymbol{P}_2+\boldsymbol{I} & \mathbf{0} \\ \boldsymbol{D}_1^T\boldsymbol{P}_2 & \boldsymbol{B}_1^T\boldsymbol{P}_1 & \mathbf{0} & -\gamma^2\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.9}$$

Proof. *Let us introduce the associated Hamiltonian as*

$$\begin{aligned} H(k,t) =&\Delta V(k,t)+y_{k+1}^T(t)y_{k+1}(t)-\gamma^2 w_{k+1}^T(t)w_{k+1}(t)\\ =&\Delta V(k,t)+\xi^T(k,t)\boldsymbol{L}^T\boldsymbol{L}\xi(k,t)-\gamma^2 w_{k+1}^T(t)w_{k+1}(t)\\ =&\begin{bmatrix} \xi^T(k,t) & w_{k+1}^T(t)\end{bmatrix}\boldsymbol{\Theta}\begin{bmatrix}\xi(k,t)\\ w_{k+1}(t)\end{bmatrix}\end{aligned} \tag{5.10}$$

where $\Delta V(k,t)$ *is defined in (4.10),* $\xi(k,t)$ *is introduced by (4.8) and*

$$\boldsymbol{\Theta}=\begin{bmatrix} \widehat{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\widehat{\boldsymbol{A}}_1+\widehat{\boldsymbol{A}}_2^T\boldsymbol{S}\widehat{\boldsymbol{A}}_2+\boldsymbol{L}^T\boldsymbol{L}-\boldsymbol{R} & \boldsymbol{P}\widehat{\boldsymbol{B}}_1+\widehat{\boldsymbol{A}}_2^T\boldsymbol{S}\widehat{\boldsymbol{D}}_1\\ \widehat{\boldsymbol{B}}_1^T\boldsymbol{P}+\widehat{\boldsymbol{D}}_1^T\boldsymbol{S}\widehat{\boldsymbol{A}}_2 & \widehat{\boldsymbol{D}}_1^T\boldsymbol{S}\widehat{\boldsymbol{D}}_1-\gamma^2\boldsymbol{I}\end{bmatrix} \tag{5.11}$$

and $\boldsymbol{P}=\operatorname{diag}(\boldsymbol{P}_1,\boldsymbol{0})$, $\boldsymbol{S}=\operatorname{diag}(\boldsymbol{P}_3,\boldsymbol{P}_2)$,

$$\widehat{\boldsymbol{B}}_1=\begin{bmatrix}\boldsymbol{B}_1\\ \boldsymbol{0}\end{bmatrix},\ \widehat{\boldsymbol{D}}_1=\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{D}_1\end{bmatrix},\ \boldsymbol{L}=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\end{bmatrix}$$

The matrix $\boldsymbol{P}_3$ *is any given positive defined matrix of appropriate dimension. Then to ensure that stability along the pass and* $\mathcal{H}_\infty$ *noise attenuation* γ *holds, (5.10) must satisfy*

$$H(k,t)\prec 0$$

Now, it is straightforward to see that the condition $\boldsymbol{\Theta}\prec 0$ *guarantees that (5.4) holds for any nonzero* $w_{k+1}(t)\in L_2^r\{[0,\infty],[0,\infty]\}$. *This implies that processes described by (5.8) are stable along the pass with the* $\mathcal{H}_\infty$ *norm less than* γ. *Finally, an obvious application of the Schur complement formula gives (5.9) and the proof is complete.* ■

Remark 5.1. *In many practice cases it is desirable to compute the minimum disturbance rejection level* γ. *This minimum can be obtained by solving a linear objective minimization problem of the following form*

$$\begin{gathered}\min_{\boldsymbol{P}_1\succ 0,\boldsymbol{P}_2\succ 0} \quad \mu\\ \text{subject to (5.9) with } \mu=\gamma^2\end{gathered} \tag{5.12}$$

5.2.2. $\mathcal{H}_\infty$ control with a static feedback controller

The purpose of this section is to study the solution to the problem of $\mathcal{H}_\infty$ disturbance attenuation in the case of full state access, i.e. the case when the control law of the form (4.17) is applied.

Under these assumptions, the following result shows that the LMI setting extends to allow the design of a control law of the form (4.17) for stability along the pass closed-loop process with a prescribed $\mathcal{H}_\infty$ disturbance attenuation.

Theorem 5.2. *Suppose that a differential LRP described by (5.2) is subject to a control law defined by (4.17). Then the resulting closed-loop process is stable along the pass and has the prescribed* $\mathcal{H}_\infty$ *disturbance attenuation* $\gamma>0$ *if there exist* *matrices $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ of compatible dimensions such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2 \\ \boldsymbol{D}_1^T & \boldsymbol{B}_1^T & \boldsymbol{0} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.13}$$

If this condition holds, the $\mathcal{H}_\infty$ controller matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are given by (4.20).

Proof. *Application of the Theorem 5.1 result shows that the closed-loop process in this case is stable along the pass with the prescribed $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$ if*

$$\begin{bmatrix} -\boldsymbol{S} & \boldsymbol{S}\overline{\boldsymbol{A}}_2 & \boldsymbol{S}\widehat{\boldsymbol{D}}_1 \\ \overline{\boldsymbol{A}}_2^T\boldsymbol{S} & \overline{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\overline{\boldsymbol{A}}_1+\boldsymbol{L}^T\boldsymbol{L}-\boldsymbol{R} & \boldsymbol{P}\widehat{\boldsymbol{B}}_1 \\ \widehat{\boldsymbol{D}}_1\boldsymbol{S} & \widehat{\boldsymbol{B}}_1^T\boldsymbol{P} & -\gamma^2\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.14}$$

where

$$\overline{\boldsymbol{A}}_1 = \begin{bmatrix} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{B}_0+\boldsymbol{B}\boldsymbol{K}_2 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \quad \overline{\boldsymbol{A}}_2 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C}+\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{D}_0+\boldsymbol{D}\boldsymbol{K}_2 \end{bmatrix}$$

Note that this last condition is not linear in $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$, $\boldsymbol{K}_1$ and $\boldsymbol{K}_1$ (it is in the BMI form). To overcome this problem, first apply the Schur complement to yield

$$\begin{bmatrix} -\boldsymbol{S} & \boldsymbol{S}\overline{\boldsymbol{A}}_2 & \boldsymbol{S}\widehat{\boldsymbol{D}}_1 & \boldsymbol{0} \\ \overline{\boldsymbol{A}}_2^T\boldsymbol{S} & \overline{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\overline{\boldsymbol{A}}_1-\boldsymbol{R} & \boldsymbol{P}\widehat{\boldsymbol{B}}_1 & \boldsymbol{L}^T \\ \widehat{\boldsymbol{D}}_1^T\boldsymbol{S} & \widehat{\boldsymbol{B}}_1^T\boldsymbol{P} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0$$

Next, substitute of $\overline{\boldsymbol{A}}_1$ and $\overline{\boldsymbol{A}}_2$ into this last expression, pre- and post-multiplying the result by $\operatorname{diag}(\boldsymbol{P}_3^{-1},\boldsymbol{P}_2^{-1},\boldsymbol{P}_1^{-1},\boldsymbol{P}_2^{-1},\boldsymbol{I},\boldsymbol{I})$ *and set $\boldsymbol{W}_1 = \boldsymbol{P}_1^{-1}$, $\boldsymbol{W}_2 = \boldsymbol{P}_2^{-1}$, $\boldsymbol{W}_3 = \boldsymbol{P}_3^{-1}$, $\boldsymbol{N}_1 = \boldsymbol{K}_1\boldsymbol{P}_1^{-1}$, $\boldsymbol{N}_2 = \boldsymbol{K}_2\boldsymbol{P}_2^{-1}$. Finally, by observing that the result does not depend on the matrix $\boldsymbol{W}_3$, the condition (5.13) is obtained. This concludes of the proof.* ■

5.2.3. $\mathcal{H}_\infty$ control with a dynamic pass profile controller

Previously, a full access to the state vector was assumed. Here, the problem of a pass profile controller existing such that the closed-loop process is stable along the pass and the $\mathcal{H}_\infty$ norm bound is less then $\gamma > 0$ is considered. It turns out that such a controller can not be designed by direct application of known methods for 2-D systems (Du and Xie, 2002) because they only deal with discrete systems. Furthermore, the intrinsic feature of LRPs is that the pass profile vector is simultaneously the output vector. Taking into account this feature, the new solution is provided.

To begin consideration of the problem, recall that the following result is used in this section.

Lemma 10. *(Dullerud and Paganini, 2000) Suppose that the $n \times n$ matrices $\boldsymbol{\Sigma} \succ 0$ and $\boldsymbol{\Gamma} \succ 0$ are given and n_c is a positive integer. Then there exists $n \times n_c$ matrices $\boldsymbol{\Sigma}_2$, $\boldsymbol{\Gamma}_2$ and $n_c \times n_c$ symmetric matrices $\boldsymbol{\Sigma}_3$, and $\boldsymbol{\Gamma}_3$, such that*

$$\begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}_2 \\ \boldsymbol{\Sigma}_2^T & \boldsymbol{\Sigma}_3 \end{bmatrix} \succ 0 \quad \text{and} \quad \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}_2 \\ \boldsymbol{\Sigma}_2^T & \boldsymbol{\Sigma}_3 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Gamma}_2 \\ \boldsymbol{\Gamma}_2^T & \boldsymbol{\Gamma}_3 \end{bmatrix}$$

if, and only if,

$$\begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{\Gamma} \end{bmatrix} \succeq 0$$

In order to solve the $\mathcal{H}_\infty$ control problem considered here, introduce the following pass profile feedback controller of the order n_c

$$\begin{aligned} \begin{bmatrix} \dot{x}_{k+1}^c(t) \\ y_{k+1}^c(t) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}_{c11} & \boldsymbol{A}_{c12} \\ \boldsymbol{A}_{c21} & \boldsymbol{A}_{c22} \end{bmatrix} \begin{bmatrix} x_{k+1}^c(t) \\ y_k^c(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_{c1} \\ \boldsymbol{B}_{c2} \end{bmatrix} y_k(t) \\ u_{k+1}(t) &= \begin{bmatrix} \boldsymbol{C}_{c1} & \boldsymbol{C}_{c2} \end{bmatrix} \begin{bmatrix} x_{k+1}^c(t) \\ y_k^c(t) \end{bmatrix} + \boldsymbol{D}_c y_k(t) \end{aligned} \tag{5.15}$$

where $x_{k+1}^c(t) \in \mathbb{R}^{n_1 \times 1}$ is the controller state vector, $y_k^c(t) \in \mathbb{R}^{m_1 \times 1}$ is the controller output vector and $n_c = n_1 + m_1$. Next, denote

$$\begin{aligned} \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix}, \ \boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{D}_1 \end{bmatrix}, \ \boldsymbol{B}_2 = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{D} \end{bmatrix}, \ \boldsymbol{C}_2 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \\ \boldsymbol{A}_c = \begin{bmatrix} \boldsymbol{A}_{c11} & \boldsymbol{A}_{c12} \\ \boldsymbol{A}_{c21} & \boldsymbol{A}_{c22} \end{bmatrix}, \ \boldsymbol{B}_c = \begin{bmatrix} \boldsymbol{B}_{c1} \\ \boldsymbol{B}_{c2} \end{bmatrix}, \ \boldsymbol{C}_c = \begin{bmatrix} \boldsymbol{C}_{c1} & \boldsymbol{C}_{c2} \end{bmatrix} \end{aligned} \tag{5.16}$$

Introduce now, the so-called augmented state and pass profile vectors

$$\dot{\overline{x}}_{k+1}(t) = \begin{bmatrix} \dot{x}_{k+1}(t) \\ \dot{x}_{k+1}^c(t) \end{bmatrix}, \ \overline{y}_k(t) = \begin{bmatrix} y_k(t) \\ y_k^c(t) \end{bmatrix}$$

and the matrices

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \ \boldsymbol{\Pi}_1 = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \ \boldsymbol{\Pi}_2 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{5.17}$$

to obtain the closed-loop process of the form

$$\begin{aligned} \begin{bmatrix} \dot{x}_{k+1}(t) \\ \overline{y}_{k+1}(t) \end{bmatrix} &= \overline{\boldsymbol{A}} \begin{bmatrix} \overline{x}_{k+1}(t) \\ \overline{y}_k(t) \end{bmatrix} + \overline{\boldsymbol{B}} w_{k+1}(t) \\ y_{k+1}(t) &= \overline{\boldsymbol{C}} \begin{bmatrix} \overline{x}_{k+1}(t) \\ \overline{y}_k(t) \end{bmatrix} \end{aligned} \tag{5.18}$$

where

$$\begin{aligned}\overline{\boldsymbol{A}}&=\boldsymbol{\Pi}\begin{bmatrix}\boldsymbol{\Phi}+\boldsymbol{B}_2\boldsymbol{D}_c\boldsymbol{C}_2 & \boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{B}_c\boldsymbol{C}_2 & \boldsymbol{A}_c\end{bmatrix}\boldsymbol{\Pi}^T=(\boldsymbol{\Pi}_1+\boldsymbol{\Pi}_2)\begin{bmatrix}\boldsymbol{\Phi}+\boldsymbol{B}_2\boldsymbol{D}_c\boldsymbol{C}_2 & \boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{B}_c\boldsymbol{C}_2 & \boldsymbol{A}_c\end{bmatrix}\boldsymbol{\Pi}^T\\ &=\boldsymbol{\Pi}_1\begin{bmatrix}\boldsymbol{\Phi}+\boldsymbol{B}_2\boldsymbol{D}_c\boldsymbol{C}_2 & \boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{B}_c\boldsymbol{C}_2 & \boldsymbol{A}_c\end{bmatrix}\boldsymbol{\Pi}^T+\boldsymbol{\Pi}_2\begin{bmatrix}\boldsymbol{\Phi}+\boldsymbol{B}_2\boldsymbol{D}_c\boldsymbol{C}_2 & \boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{B}_c\boldsymbol{C}_2 & \boldsymbol{A}_c\end{bmatrix}\boldsymbol{\Pi}^T=\overline{\boldsymbol{A}}_1+\overline{\boldsymbol{A}}_2\\ \overline{\boldsymbol{B}}&=\boldsymbol{\Pi}_1\begin{bmatrix}\boldsymbol{\Omega}\\ \boldsymbol{0}\end{bmatrix}+\boldsymbol{\Pi}_2\begin{bmatrix}\boldsymbol{\Omega}\\ \boldsymbol{0}\end{bmatrix}=\overline{\boldsymbol{B}}_1+\overline{\boldsymbol{B}}_2,\ \overline{\boldsymbol{C}}=\begin{bmatrix}\boldsymbol{C}_2 & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Pi}^T\end{aligned}$$

In the following, introduce the matrix of controller data as

$$\boldsymbol{\Theta}=\begin{bmatrix}\boldsymbol{D}_c & \boldsymbol{C}_c\\ \boldsymbol{B}_c & \boldsymbol{A}_c\end{bmatrix} \tag{5.19}$$

and

$$\begin{aligned}&\boldsymbol{\mathcal{A}}_1=\boldsymbol{\Pi}_1\begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Pi},\ \boldsymbol{\mathcal{A}}_2=\boldsymbol{\Pi}_2\begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Pi},\ \boldsymbol{\Gamma}_1=\boldsymbol{\Pi}_1\begin{bmatrix}\boldsymbol{B}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix},\ \boldsymbol{\Gamma}_2=\boldsymbol{\Pi}_2\begin{bmatrix}\boldsymbol{B}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix},\\ &\boldsymbol{\mathcal{C}}_2=\begin{bmatrix}\boldsymbol{C}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\boldsymbol{\Pi}^T,\ \boldsymbol{\mathcal{C}}=\begin{bmatrix}\boldsymbol{C}_2 & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Pi}^T,\ \boldsymbol{\mathcal{B}}_1=\boldsymbol{\Pi}_1\begin{bmatrix}\boldsymbol{\Omega}\\ \boldsymbol{0}\end{bmatrix},\ \boldsymbol{\mathcal{B}}_2=\boldsymbol{\Pi}_2\begin{bmatrix}\boldsymbol{\Omega}\\ \boldsymbol{0}\end{bmatrix}\end{aligned} \tag{5.20}$$

It allows presenting the closed-loop process matrices in the form affine in the controller data matrix $\boldsymbol{\Theta}$ as

$$\overline{\boldsymbol{A}}_1=\boldsymbol{\mathcal{A}}_1+\boldsymbol{\Gamma}_1\boldsymbol{\Theta}\boldsymbol{\mathcal{C}}_2,\ \overline{\boldsymbol{A}}_2=\boldsymbol{\mathcal{A}}_2+\boldsymbol{\Gamma}_2\boldsymbol{\Theta}\boldsymbol{\mathcal{C}}_2,\ \overline{\boldsymbol{B}}_1=\boldsymbol{\mathcal{B}}_1,\ \overline{\boldsymbol{B}}_2=\boldsymbol{\mathcal{B}}_2,\overline{\boldsymbol{C}}=\boldsymbol{\mathcal{C}} \tag{5.21}$$

Hence, we have the following result based on Theorem 5.1.

Theorem 5.3. *A differential LRP described by (5.2) is stable along the pass and has the $\mathcal{H}_\infty$ disturbance attenuation $\gamma>0$ if there exist matrices $\boldsymbol{S}_v\succ 0$, $\boldsymbol{P}_h\succ 0$, such that the following inequality holds*

$$\begin{bmatrix}-\boldsymbol{S} & \boldsymbol{S}\overline{\boldsymbol{A}}_2 & \boldsymbol{S}\overline{\boldsymbol{B}}_2 & \boldsymbol{0}\\ \overline{\boldsymbol{A}}_2^T\boldsymbol{S} & \overline{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\overline{\boldsymbol{A}}_1-\boldsymbol{R} & \boldsymbol{P}\overline{\boldsymbol{B}}_1 & \overline{\boldsymbol{C}}^T\\ \overline{\boldsymbol{B}}_2^T\boldsymbol{S} & \overline{\boldsymbol{B}}_1^T\boldsymbol{P} & -\gamma^2\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \overline{\boldsymbol{C}} & \boldsymbol{0} & -\boldsymbol{I}\end{bmatrix}\prec 0 \tag{5.22}$$

where $\boldsymbol{S}=\operatorname{diag}(\boldsymbol{I},\boldsymbol{S}_v)$, $\boldsymbol{P}=\operatorname{diag}(\boldsymbol{P}_h,\boldsymbol{I})$, $\boldsymbol{R}=\operatorname{diag}(\boldsymbol{0},\boldsymbol{S}_v)$.

This can be resolved to the following form.

Theorem 5.4. *If there exist matrices $\boldsymbol{P}_{h_{11}}\succ 0$, $\boldsymbol{U}_{h_{11}}\succ 0$, $\boldsymbol{S}_{v_{11}}\succ 0$, $\boldsymbol{T}_{v_{11}}\succ 0$ such that the LMIs defined by (5.23)–(5.25) hold then there exists a controller of the form (5.15) which guarantees that the differential LRP described by (5.18) is* *stable along the pass and has the $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$.*

$$\begin{bmatrix} \boldsymbol{\mathcal{W}}_c^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} -\boldsymbol{T}_{v_{11}}+\boldsymbol{D}_0\boldsymbol{T}_{v_{11}}\boldsymbol{D}_0^T & \boldsymbol{C}\boldsymbol{U}_{h_{11}}+\boldsymbol{D}_0\boldsymbol{T}_{v_{11}}\boldsymbol{B}_0^T & \boldsymbol{D}_1 & \boldsymbol{D}_0\boldsymbol{T}_{v_{11}} \\ \boldsymbol{U}_{h_{11}}\boldsymbol{C}^T+\boldsymbol{B}_0\boldsymbol{T}_{v_{11}}\boldsymbol{D}_0^T & \boldsymbol{U}_{h_{11}}\boldsymbol{A}^T+\boldsymbol{A}\boldsymbol{U}_{h_{11}}+\boldsymbol{B}_0\boldsymbol{T}_{v_{11}}\boldsymbol{B}_0^T & \boldsymbol{B}_1 & \boldsymbol{B}_0\boldsymbol{T}_{v_{11}} \\ \boldsymbol{D}_1^T & \boldsymbol{B}_1^T & -\gamma^2\boldsymbol{I} & \mathbf{0} \\ \boldsymbol{T}_{v_{11}}\boldsymbol{D}_0^T & \boldsymbol{T}_{v_{11}}\boldsymbol{B}_0^T & \mathbf{0} & -\boldsymbol{I}+\boldsymbol{T}_{v_{11}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{W}}_c & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.23}$$

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}_{h_{11}}+\boldsymbol{P}_{h_{11}}\boldsymbol{A}+\boldsymbol{C}^T\boldsymbol{S}_{v_{11}}\boldsymbol{C} & \boldsymbol{P}_{h_{11}}\boldsymbol{B}_1+\boldsymbol{C}^T\boldsymbol{S}_{v_{11}}\boldsymbol{D}_1 \\ \boldsymbol{B}_1^T\boldsymbol{P}_{h_{11}}+\boldsymbol{D}_1^T\boldsymbol{S}_{v_{11}}\boldsymbol{C} & \boldsymbol{D}_1^T\boldsymbol{S}_{v_{11}}\boldsymbol{D}_1-\gamma^2\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.24}$$

$$\begin{bmatrix} \boldsymbol{P}_{h11} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{U}_{h11} \end{bmatrix} \succeq 0, \quad \begin{bmatrix} \boldsymbol{S}_{v11} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{T}_{v11} \end{bmatrix} \succeq 0 \tag{5.25}$$

where $\boldsymbol{\mathcal{W}}_c$ is full column rank matrix whose image satisfies

$$\mathrm{Im}\,(\boldsymbol{\mathcal{W}}_c) = \ker\left(\begin{bmatrix} \boldsymbol{D}^T \\ \boldsymbol{B}^T \end{bmatrix}\right)$$

Suppose now that the LMIs (5.23)-(5.25) are feasible. Then the following is a systematic procedure for obtaining the corresponding controller state-space matrices.

Step 1: Compute the matrices $\boldsymbol{P}_{h12}$, $\boldsymbol{P}_{v12}$ using the following formulas

$$\boldsymbol{S}_{v_{12}}\boldsymbol{S}_{v_{22}}^{-1}\boldsymbol{S}_{v_{12}}^T = \boldsymbol{S}_{v_{11}} - \boldsymbol{T}_{v_{11}}^{-1}$$
$$\boldsymbol{P}_{h_{12}}\boldsymbol{P}_{h_{22}}^{-1}\boldsymbol{P}_{h_{12}}^T = \boldsymbol{P}_{h_{11}} - \boldsymbol{U}_{h_{11}}^{-1}$$

where $\boldsymbol{P}_{h22} = \boldsymbol{I}$ and $\boldsymbol{S}_{v22} = \boldsymbol{I}$

Step 2: Construct the matrices $\boldsymbol{P}_h \succ 0$ and $\boldsymbol{S}_v \succ 0$ as

$$\boldsymbol{P}_h = \begin{bmatrix} \boldsymbol{P}_{h_{11}} & \boldsymbol{P}_{h_{12}}^T \\ \boldsymbol{P}_{h_{12}} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{S}_v = \begin{bmatrix} \boldsymbol{S}_{v_{11}} & \boldsymbol{S}_{v_{12}}^T \\ \boldsymbol{S}_{v_{12}} & \boldsymbol{I} \end{bmatrix}$$

and then the matrices $\boldsymbol{S} = \mathrm{diag}(\boldsymbol{I}, \boldsymbol{S}_v)$, $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{P}_h, \boldsymbol{I})$, $\boldsymbol{R} = \mathrm{diag}(\mathbf{0}, \boldsymbol{S}_v)$

Step 3: Compute the matrices $\boldsymbol{M}$, $\boldsymbol{N}$ and $\boldsymbol{\Psi}$ defined in (5.28)

Step 4: Solve the following LMI

$$\boldsymbol{\Psi} + \boldsymbol{M}^T\boldsymbol{\Theta}\boldsymbol{N} + \boldsymbol{N}^T\boldsymbol{\Theta}^T\boldsymbol{M} \prec 0$$

to obtain (5.19) i.e. the matrices which define the controller state-space model (5.15).

Proof. *Using matrix definitions (5.21), the inequality (5.22) can be rewritten as*

$$\begin{bmatrix} -\boldsymbol{S} & \boldsymbol{S}\boldsymbol{\mathcal{A}}_2+\boldsymbol{S}\boldsymbol{\Gamma}_2\boldsymbol{\Theta}\boldsymbol{\mathcal{C}}_2 & \boldsymbol{S}\boldsymbol{\mathcal{B}}_2 & \boldsymbol{0} \\ \boldsymbol{\mathcal{A}}_2^T\boldsymbol{S}+\boldsymbol{\mathcal{C}}_2^T\boldsymbol{\Theta}^T\boldsymbol{\Gamma}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{A}}_1^T\boldsymbol{P}+\boldsymbol{\mathcal{C}}_2^T\boldsymbol{\Theta}^T\boldsymbol{\Gamma}_1^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{\mathcal{A}}_1+\boldsymbol{P}\boldsymbol{\Gamma}_1\boldsymbol{\Theta}\boldsymbol{\mathcal{C}}_2-\boldsymbol{R} & \boldsymbol{P}\boldsymbol{\mathcal{B}}_1 & \boldsymbol{\mathcal{C}}^T \\ \boldsymbol{\mathcal{B}}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{B}}_1^T\boldsymbol{P} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{C}} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0$$

$$\Leftrightarrow \begin{bmatrix} -\boldsymbol{S} & \boldsymbol{S}\boldsymbol{\mathcal{A}}_2 & \boldsymbol{S}\boldsymbol{\mathcal{B}}_2 & \boldsymbol{0} \\ \boldsymbol{\mathcal{A}}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{\mathcal{A}}_1-\boldsymbol{R} & \boldsymbol{P}\boldsymbol{\mathcal{B}}_1 & \boldsymbol{\mathcal{C}}^T \\ \boldsymbol{\mathcal{B}}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{B}}_1^T\boldsymbol{P} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{C}} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{S}\boldsymbol{\Gamma}_2\boldsymbol{\Theta}\boldsymbol{\mathcal{C}}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}\boldsymbol{\Gamma}_1\boldsymbol{\Theta}\boldsymbol{\mathcal{C}}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$
$$+ \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{C}}_2^T\boldsymbol{\Theta}^T\boldsymbol{\Gamma}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{C}}_2^T\boldsymbol{\Theta}^T\boldsymbol{\Gamma}_1^T\boldsymbol{P} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \prec 0 \tag{5.26}$$

It is straightforward to see that (5.26) can be rewritten as

$$\begin{bmatrix} -\boldsymbol{S} & \boldsymbol{S}\boldsymbol{\mathcal{A}}_2 & \boldsymbol{S}\boldsymbol{\mathcal{B}}_2 & \boldsymbol{0} \\ \boldsymbol{\mathcal{A}}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{\mathcal{A}}_1-\boldsymbol{R} & \boldsymbol{P}\boldsymbol{\mathcal{B}}_1 & \boldsymbol{\mathcal{C}}^T \\ \boldsymbol{\mathcal{B}}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{B}}_1^T\boldsymbol{P} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{C}} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{S}\boldsymbol{\Gamma}_2 \\ \boldsymbol{P}\boldsymbol{\Gamma}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\Theta} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\mathcal{C}}_2 & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$
$$+ \begin{bmatrix} \boldsymbol{\Gamma}_2^T\boldsymbol{S} & \boldsymbol{\Gamma}_1^T\boldsymbol{P} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{\Theta}^T \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\mathcal{C}}_2^T \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \prec 0 \tag{5.27}$$

Now, define the matrices

$$\boldsymbol{\Psi} = \begin{bmatrix} -\boldsymbol{S} & \boldsymbol{S}\boldsymbol{\mathcal{A}}_2 & \boldsymbol{S}\boldsymbol{\mathcal{B}}_2 & \boldsymbol{0} \\ \boldsymbol{\mathcal{A}}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{\mathcal{A}}_1-\boldsymbol{R} & \boldsymbol{P}\boldsymbol{\mathcal{B}}_1 & \boldsymbol{\mathcal{C}}^T \\ \boldsymbol{\mathcal{B}}_2^T\boldsymbol{S} & \boldsymbol{\mathcal{B}}_1^T\boldsymbol{P} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{C}} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{M}^T = \begin{bmatrix} \boldsymbol{S}\boldsymbol{\Gamma}_2 \\ \boldsymbol{P}\boldsymbol{\Gamma}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \tag{5.28}$$
$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\mathcal{C}}_2 & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

to present (5.27) in the form

$$\boldsymbol{\Psi} + \boldsymbol{M}^T\boldsymbol{\Theta}\boldsymbol{N} + \boldsymbol{N}^T\boldsymbol{\Theta}^T\boldsymbol{M} \prec 0 \tag{5.29}$$

In the sequel, invoke Lemma 9 to eliminate the matrix variable $\boldsymbol{\Theta}$ *and obtain*

$$\boldsymbol{\mathcal{W}}_M^T\boldsymbol{\Psi}\boldsymbol{\mathcal{W}}_M \prec 0 \ \text{ and } \ \boldsymbol{\mathcal{W}}_N^T\boldsymbol{\Psi}\boldsymbol{\mathcal{W}}_N \prec 0 \tag{5.30}$$

where

$$\boldsymbol{\mathcal{W}}_M \in \ker(\boldsymbol{M}), \ \ \boldsymbol{\mathcal{W}}_N \in \ker(\boldsymbol{N}) \tag{5.31}$$

Next, note that

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{\Gamma}_2^T \boldsymbol{S} \; \boldsymbol{\Gamma}_1^T \boldsymbol{P} \; \boldsymbol{0} \; \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma}_2^T \; \boldsymbol{\Gamma}_1^T \; \boldsymbol{0} \; \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = \boldsymbol{M}_n \boldsymbol{V} \tag{5.32}$$

where the matrices $\boldsymbol{P}$ *and* $\boldsymbol{S}$ *are partitioned as*

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{h_{11}} & \boldsymbol{P}_{h_{12}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{P}_{h_{12}}^T & \boldsymbol{P}_{h_{22}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{S}_{v_{11}} & \boldsymbol{S}_{v_{12}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{S}_{v_{12}}^T & \boldsymbol{S}_{v_{22}} \end{bmatrix}$$

and

$$\boldsymbol{P}_h = \begin{bmatrix} \boldsymbol{P}_{h_{11}} & \boldsymbol{P}_{h_{12}} \\ \boldsymbol{P}_{h_{12}}^T & \boldsymbol{P}_{h_{22}} \end{bmatrix}, \; \boldsymbol{P}_v = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \; \boldsymbol{S}_h = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \; \boldsymbol{S}_v = \begin{bmatrix} \boldsymbol{S}_{v_{11}} & \boldsymbol{S}_{v_{12}} \\ \boldsymbol{S}_{v_{12}}^T & \boldsymbol{S}_{v_{22}} \end{bmatrix}$$

Observe that the solution of (5.22) is independent of the matrices $\boldsymbol{P}_v$ *and* $\boldsymbol{S}_h$ *but they are necessary to compute* $\boldsymbol{V}^{-1}$ *(the matrix* $\boldsymbol{V}$ *is defined in (5.32)). Furthermore, it is important now to observe that*

$$\ker(\boldsymbol{M}_n \boldsymbol{V}) = \boldsymbol{V}^{-1} \ker(\boldsymbol{M}_n)$$

and using (5.31) yields that

$$\boldsymbol{\mathcal{W}}_M = \boldsymbol{V}^{-1} \boldsymbol{\mathcal{W}}_{M_n}.$$

Therefore, the first inequality in (5.30) is

$$\boldsymbol{\mathcal{W}}_{M_n}^T \boldsymbol{V}^{-T} \boldsymbol{\Psi} \boldsymbol{V}^{-1} \boldsymbol{\mathcal{W}}_{M_n} \prec 0$$

Furthermore, introduce the matrix $\boldsymbol{\Xi}$ *as*

$$\boldsymbol{\Xi} = \boldsymbol{V}^{-T} \boldsymbol{\Psi} \boldsymbol{V}^{-1} = \begin{bmatrix} -\boldsymbol{S}^{-1} & \boldsymbol{\mathcal{A}}_2 \boldsymbol{P}^{-1} & \boldsymbol{\mathcal{B}}_2 & \boldsymbol{0} \\ \boldsymbol{P}^{-1} \boldsymbol{\mathcal{A}}_2^T & \boldsymbol{P}^{-1} \boldsymbol{\mathcal{A}}_1^T + \boldsymbol{\mathcal{A}}_1 \boldsymbol{P}^{-1} - \boldsymbol{R} & \boldsymbol{\mathcal{B}}_1 & \boldsymbol{P}^{-1} \boldsymbol{\mathcal{C}}^T \\ \boldsymbol{\mathcal{B}}_2^T & \boldsymbol{\mathcal{B}}_1^T & -\gamma^2 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{C}} \boldsymbol{P}^{-1} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}$$

Then, two inequalities of (5.30) are

$$\boldsymbol{\mathcal{W}}_{M_n}^T \boldsymbol{\Xi} \boldsymbol{\mathcal{W}}_{M_n} \prec 0 \tag{5.33}$$

and

$$\boldsymbol{\mathcal{W}}_N^T \boldsymbol{\Psi} \boldsymbol{\mathcal{W}}_N \prec 0 \tag{5.34}$$

These inequalities clearly are in the BMI form, thus to obtain LMI for (5.33) and (5.34) some matrix manipulations are required. First, rewrite matrices in the full form

$$\boldsymbol{M}_n = \left[\; \boldsymbol{\Gamma_2}^T \; \middle| \; \boldsymbol{\Gamma_1}^T \; \middle| \; \boldsymbol{0} \; \middle| \; \boldsymbol{0} \; \right] = \left[\begin{array}{cccc|cccc|c|c} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{array} \right]$$

$$\boldsymbol{\Gamma}_1=\boldsymbol{\Pi}_1\begin{bmatrix}\boldsymbol{B}_2&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{B}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{D}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}=\begin{bmatrix}\boldsymbol{B}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\end{bmatrix}$$

$$\boldsymbol{\Gamma}_2=\boldsymbol{\Pi}_2\begin{bmatrix}\boldsymbol{B}_2&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{B}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{D}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{D}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}$$

$$\boldsymbol{N}=\left[\begin{array}{c|c|c|c}\boldsymbol{0}&\boldsymbol{\mathcal{C}}_2&\boldsymbol{0}&\boldsymbol{0}\end{array}\right]=\left[\begin{array}{cccc|cccc|c|c}\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\end{array}\right]$$

$$\boldsymbol{\mathcal{C}}_2=\begin{bmatrix}\boldsymbol{C}_2&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}\boldsymbol{\Pi}^T=\begin{bmatrix}\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}$$

It is clear to see that the kernels of $\boldsymbol{M}_n$ and $\boldsymbol{N}$ are images of

$$\boldsymbol{\mathcal{W}}_{M_n}=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{\mathcal{N}}_D&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{\mathcal{N}}_B&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix},\ \boldsymbol{\mathcal{W}}_N=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix}$$

where

$$\boldsymbol{\mathcal{N}}_B\in\ker(\boldsymbol{B}^T),\ \ \boldsymbol{\mathcal{N}}_D\in\ker(\boldsymbol{D}^T),$$

Moreover, the matrices $\boldsymbol{P}^{-1}$, $\boldsymbol{S}^{-1}$ and $\boldsymbol{R}$ can be partitioned in the following way

$$\boldsymbol{P}^{-1}=\begin{bmatrix}\boldsymbol{U}_{h_{11}}&\boldsymbol{U}_{h_{12}}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{U}_{h_{12}}^T&\boldsymbol{U}_{h_{22}}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{I}\end{bmatrix},\ \boldsymbol{S}^{-1}=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{T}_{v_{11}}&\boldsymbol{T}_{v_{12}}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{T}_{v_{12}}^T&\boldsymbol{T}_{v_{22}}\end{bmatrix},\ \boldsymbol{R}=\begin{bmatrix}\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{S}_{v_{11}}&\boldsymbol{S}_{v_{12}}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{S}_{v_{12}}^T&\boldsymbol{S}_{v_{22}}\end{bmatrix}$$

It allows us to rewrite the matrix $\boldsymbol{\Xi}$ *as*

$$\boldsymbol{\Xi}=\left[\begin{array}{cccc|cccc|c|c}
-\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & -\boldsymbol{T}_{v_{11}} & -\boldsymbol{T}_{v_{12}} & \boldsymbol{C}\boldsymbol{U}_{h_{11}} & \boldsymbol{C}\boldsymbol{U}_{h_{12}} & \boldsymbol{D}_0 & \mathbf{0} & \boldsymbol{D}_1 & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & -\boldsymbol{T}_{v_{12}}^T & -\boldsymbol{T}_{v_{22}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\hline
\mathbf{0} & \mathbf{0} & \boldsymbol{U}_{h_{11}}\boldsymbol{C}^T & \mathbf{0} & \boldsymbol{U}_{h_{11}}\boldsymbol{A}^T+\boldsymbol{A}\boldsymbol{U}_{h_{11}} & \boldsymbol{A}\boldsymbol{U}_{h_{12}} & \boldsymbol{B}_0 & \mathbf{0} & \boldsymbol{B}_1 & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \boldsymbol{U}_{h_{12}}^T\boldsymbol{C}^T & \mathbf{0} & \boldsymbol{U}_{h_{12}}^T\boldsymbol{A}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \boldsymbol{D}_0^T & \mathbf{0} & \boldsymbol{B}_0^T & \mathbf{0} & -\boldsymbol{S}_{v_{11}} & -\boldsymbol{S}_{v_{12}} & \mathbf{0} & \boldsymbol{I}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{S}_{v_{12}}^T & -\boldsymbol{S}_{v_{22}} & \mathbf{0} & \mathbf{0}\\
\hline
\mathbf{0} & \mathbf{0} & \boldsymbol{D}_1^T & \mathbf{0} & \boldsymbol{B}_1^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\gamma^2\boldsymbol{I} & \mathbf{0}\\
\hline
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & -\boldsymbol{I}
\end{array}\right]$$

Consider the first BMI constraint (5.34)

$$\boldsymbol{\mathcal{W}}_{M_n}^T\boldsymbol{\Xi}\boldsymbol{\mathcal{W}}_{M_n}\prec 0$$

By observing that the forth and sixth rows of $\boldsymbol{\mathcal{W}}_{M_n}$ *are zero, the last inequality can be rewritten as*

$$\begin{bmatrix}
\boldsymbol{\mathcal{N}}_D^T & \boldsymbol{\mathcal{N}}_B^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}
\end{bmatrix}
\left[\begin{array}{cc|cc|cc}
-\boldsymbol{T}_{v_{11}} & \boldsymbol{C}\boldsymbol{U}_{h_{11}} & \boldsymbol{D}_0 & \mathbf{0} & \boldsymbol{D}_1 & \mathbf{0}\\
\boldsymbol{U}_{h_{11}}\boldsymbol{C}^T & \boldsymbol{U}_{h_{11}}\boldsymbol{A}^T+\boldsymbol{A}\boldsymbol{U}_{h_{11}} & \boldsymbol{B}_0 & \mathbf{0} & \boldsymbol{B}_1 & \mathbf{0}\\
\hline
\boldsymbol{D}_0^T & \boldsymbol{B}_0^T & -\boldsymbol{S}_{v_{11}} & -\boldsymbol{S}_{v_{12}} & \mathbf{0} & \boldsymbol{I}\\
\mathbf{0} & \mathbf{0} & -\boldsymbol{S}_{v_{12}}^T & -\boldsymbol{S}_{v_{22}} & \mathbf{0} & \mathbf{0}\\
\hline
\boldsymbol{D}_1^T & \boldsymbol{B}_1^T & \mathbf{0} & \mathbf{0} & -\gamma^2\boldsymbol{I} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & -\boldsymbol{I}
\end{array}\right]$$

$$\times\begin{bmatrix}
\boldsymbol{\mathcal{N}}_D & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\boldsymbol{\mathcal{N}}_B & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}
\end{bmatrix}\prec 0$$

Next, application of the Schur complement formula yields (5.23). In order to obtain the inequality (5.24), rewrite the matrix $\boldsymbol{\Psi}$ *as*

$$\boldsymbol{\Psi}=\begin{bmatrix}
-\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & -\boldsymbol{S}_{v_{11}} & -\boldsymbol{S}_{v_{12}} & \boldsymbol{S}_{v_{11}}\boldsymbol{C} & \mathbf{0} & \boldsymbol{S}_{v_{11}}\boldsymbol{D}_0 & \mathbf{0} & \boldsymbol{S}_{v_{11}}\boldsymbol{D}_1 & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & -\boldsymbol{S}_{v_{12}}^T & -\boldsymbol{S}_{v_{22}} & \boldsymbol{S}_{v_{12}}^T\boldsymbol{C} & \mathbf{0} & \boldsymbol{S}_{v_{12}}^T\boldsymbol{D}_0 & \mathbf{0} & \boldsymbol{S}_{v_{12}}^T\boldsymbol{D}_1 & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \boldsymbol{C}^T\boldsymbol{S}_{v_{11}} & \boldsymbol{C}^T\boldsymbol{S}_{v_{12}} & \boldsymbol{A}^T\boldsymbol{P}_{h_{11}}+\boldsymbol{P}_{h_{11}}\boldsymbol{A} & \boldsymbol{A}\boldsymbol{P}_{h_{12}} & \boldsymbol{P}_{h_{11}}\boldsymbol{B}_0 & \mathbf{0} & \boldsymbol{P}_{h_{11}}\boldsymbol{B}_1 & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{P}_{h_{12}}^T\boldsymbol{A} & \mathbf{0} & \boldsymbol{P}_{h_{12}}^T\boldsymbol{B}_0 & \mathbf{0} & \boldsymbol{P}_{h_{12}}^T\boldsymbol{B}_1 & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \boldsymbol{D}_0^T\boldsymbol{S}_{v_{11}} & \boldsymbol{D}_0^T\boldsymbol{S}_{v_{11}} & \boldsymbol{B}_0^T\boldsymbol{P}_{h_{11}} & \boldsymbol{B}_0^T\boldsymbol{P}_{h_{12}} & -\boldsymbol{S}_{v_{11}} & -\boldsymbol{S}_{v_{12}} & \mathbf{0} & \boldsymbol{I}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{S}_{v_{12}}^T & -\boldsymbol{S}_{v_{22}} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \boldsymbol{D}_1^T\boldsymbol{S}_{v_{11}} & \boldsymbol{D}_1^T\boldsymbol{S}_{v_{12}} & \boldsymbol{B}_1^T\boldsymbol{P}_{h_{11}} & \boldsymbol{B}_1^T\boldsymbol{P}_{h_{12}} & \mathbf{0} & \mathbf{0} & -\gamma^2\boldsymbol{I} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & -\boldsymbol{I}
\end{bmatrix}$$

Next, by observing that the some rows of $\boldsymbol{\mathcal{W}}_N$ *are zero, the condition (5.34)* *reduces to*

$$\begin{bmatrix} -\boldsymbol{S}_{v_{11}} & -\boldsymbol{S}_{v_{12}} & \boldsymbol{S}_{v_{11}}\boldsymbol{C} & \boldsymbol{S}_{v_{11}}\boldsymbol{D}_1 & \boldsymbol{0} \\ -\boldsymbol{S}_{v_{12}}^T & -\boldsymbol{S}_{v_{22}} & \boldsymbol{S}_{v_{12}}^T\boldsymbol{C} & \boldsymbol{S}_{v_{12}}^T\boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{C}^T\boldsymbol{S}_{v_{11}} & \boldsymbol{C}^T\boldsymbol{S}_{v_{12}} & \boldsymbol{A}^T\boldsymbol{P}_{h_{11}}+\boldsymbol{P}_{h_{11}}\boldsymbol{A} & \boldsymbol{P}_{h_{11}}\boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{D}_1^T\boldsymbol{S}_{v_{11}} & \boldsymbol{D}_1^T\boldsymbol{S}_{v_{12}} & \boldsymbol{B}_1^T\boldsymbol{P}_{h_{11}} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0$$

Further, application of the Schur complement formula gives

$$\left(\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}_{h_{11}}+\boldsymbol{P}_{h_{11}}\boldsymbol{A} & \boldsymbol{P}_{h_{11}}\boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{B}_1^T\boldsymbol{P}_{h_{11}} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}^T\boldsymbol{S}_{v_{11}} & \boldsymbol{C}^T\boldsymbol{S}_{v_{12}} \\ \boldsymbol{D}_1^T\boldsymbol{S}_{v_{11}} & \boldsymbol{D}_1^T\boldsymbol{S}_{v_{12}} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{T}_{v_{11}} & \boldsymbol{T}_{v_{12}} \\ \boldsymbol{T}_{v_{12}}^T & \boldsymbol{T}_{v_{22}} \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_{v_{11}}\boldsymbol{C} & \boldsymbol{S}_{v_{11}}\boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{S}_{v_{12}}^T\boldsymbol{C} & \boldsymbol{S}_{v_{12}}^T\boldsymbol{D}_1 & \boldsymbol{0} \end{bmatrix}\right) \prec 0$$

which is equivalent to LMI (5.24). Additionally, applying of Lemma 10 gives as the result the conditions defined in (5.25) and the proof is completed. ■

The disturbance attenuation level γ can be optimized by solving the following linear minimization problem

$$\begin{aligned} &\min_{\boldsymbol{P}_{h_{11}}\succ 0, \boldsymbol{U}_{h_{11}}\succ 0, \boldsymbol{S}_{v_{11}}\succ 0, \boldsymbol{T}_{v_{11}}\succ 0} \quad \mu \\ &\text{subject to (5.23)} - \text{(5.25) with } \mu = \gamma^2 \end{aligned} \tag{5.35}$$

To show the usefulness of this result, let us consider the following numerical example.

Example 5.1. *Consider the differential LRP of the form (5.2) with matrices given by*

$$\boldsymbol{A} = \begin{bmatrix} -1.4684 & -1.3745 \\ -1.4080 & -1.6290 \end{bmatrix}, \boldsymbol{B}_0 = \begin{bmatrix} 0.0309 & 0.2553 \\ 0.0635 & 0.2229 \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} 0.0810 & 0.0948 \\ 0.0610 & 0.0289 \end{bmatrix},$$

$$\boldsymbol{D}_0 = \begin{bmatrix} 0.0494 & 0.0510 \\ 0.0530 & 0.0227 \end{bmatrix}, \boldsymbol{B}_1 = \begin{bmatrix} 1.9635 & 0.9644 \\ 1.0872 & 1.6311 \end{bmatrix}, \boldsymbol{D}_1 = \begin{bmatrix} 1.5410 & 0.1196 \\ 1.2650 & 0.1512 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 1.9070 & 1.2824 \\ 2.6290 & 2.1252 \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} 0.2106 & 1.5249 \\ 0.0572 & 2.6082 \end{bmatrix}$$

With this data, application of the controller design procedure of (5.35) yields

$$\boldsymbol{P}_{h11} = \begin{bmatrix} 0.1169 & 0.1036 \\ 0.1036 & 0.1740 \end{bmatrix}, \boldsymbol{U}_{h11} = 10^6 \begin{bmatrix} 8.3795 & 0.5142 \\ 0.5142 & 9.1918 \end{bmatrix},$$

$$\boldsymbol{S}_{v11} = \begin{bmatrix} 6.1931 & -5.8899 \\ -5.8899 & 7.6801 \end{bmatrix}, \boldsymbol{T}_{v11} = \begin{bmatrix} 0.8599 & 0.1589 \\ 0.1589 & 0.8198 \end{bmatrix}$$

and the corresponding $\mathcal{H}_\infty$ controller matrices are

$$\boldsymbol{A}_c = \begin{bmatrix} -3.8573 & -1.5123 \\ -1.7765 & -0.9105 \end{bmatrix}, \boldsymbol{B}_c = \begin{bmatrix} 0.1405 & 0.1581 \\ 0.0712 & 0.0991 \end{bmatrix},$$

$$\boldsymbol{C}_c = \begin{bmatrix} 0.7487 & 0.7430 \\ 0.0520 & -0.0499 \end{bmatrix}, \boldsymbol{D}_c = \begin{bmatrix} -0.1041 & -0.2130 \\ -0.0180 & -0.0040 \end{bmatrix}$$

Moreover, the minimum γ derived in this example is 2.1530.

5.2.4. Numerical computations of the Lyapunov matrix

It should be pointed out that providing the Lyapunov matrix results in computational problems which are nontrivial due to required of existence of the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ such that $\boldsymbol{Y} = \boldsymbol{X}^{-1}$. Unfortunately, this problem is non-convex therefore the convex formulation is required. Moreover, the controller order is equal to the rank of $\boldsymbol{X} - \boldsymbol{Y}$ so it is hard to specify the controller order. It is shown in (Fu and Luo, 1997) that computational problems which have arisen in fixed output feedback design are $\mathcal{NP}$-hard problems.

In the above design procedure, the dynamic controller with unspecified order was considered. In this case, the constraint $\boldsymbol{Y} = \boldsymbol{X}^{-1} \succ 0$ can be replaced by $\boldsymbol{Y} \succeq \boldsymbol{X}^{-1} \succ 0$ (Iwasaki and Skelton, 1994) which, after applying the Schur complement formula, leads to

$$\begin{bmatrix} \boldsymbol{Y} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{X} \end{bmatrix} \succeq 0$$

Thus, the convex formulation has been found and it can be applied in $\mathcal{H}_\infty$ controller design - see conditions in (5.25). Unfortunately, since one of the block $\boldsymbol{X}$ is chosen as $\boldsymbol{I}$ (the identity matrix) then the condition number of $\boldsymbol{X}$ can be large. To avoid computation problems, the approach based on the singular value decomposition (SVD) is used. As the first step, SVD of $\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$ is computed

$$\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

Next, the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ are chosen as

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}^{\frac{1}{2}}$$
$$\boldsymbol{N} = \boldsymbol{V}\boldsymbol{\Sigma}^{\frac{1}{2}}$$

and they satisfy $\boldsymbol{M}\boldsymbol{N}^T = \boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$. This allows us to compute the Lyapunov matrix $\boldsymbol{L}$ from the relation

$$\begin{bmatrix} \boldsymbol{Y} & \boldsymbol{I} \\ \boldsymbol{N}^T & \boldsymbol{0} \end{bmatrix} = \boldsymbol{L} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{M}^T \end{bmatrix}$$

As the next step, partition $\boldsymbol{L}$ as

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{L}_{12} \\ \boldsymbol{L}_{12}^T & \boldsymbol{L}_{22} \end{bmatrix}$$

to obtain

$$\boldsymbol{L}\begin{bmatrix} \boldsymbol{I} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{M}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{L}_{12} \\ \boldsymbol{L}_{12}^T & \boldsymbol{L}_{22} \end{bmatrix}\begin{bmatrix} \boldsymbol{I} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{M}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{L}_{11}\boldsymbol{X} + \boldsymbol{L}_{12}\boldsymbol{M}^T \\ \boldsymbol{L}_{12}^T & \boldsymbol{L}_{12}^T\boldsymbol{X} + \boldsymbol{L}_{22}\boldsymbol{M}^T \end{bmatrix}$$

Based on the above equation, the blocks of $\boldsymbol{L}$ can be computed from

$$\begin{aligned} \boldsymbol{L}_{11} &= \boldsymbol{Y} \\ \boldsymbol{L}_{12}^T &= \boldsymbol{N}^T \\ \boldsymbol{L}_{11}\boldsymbol{X} + \boldsymbol{L}_{12}\boldsymbol{M}^T &= \boldsymbol{I} \\ \boldsymbol{L}_{12}^T\boldsymbol{X} + \boldsymbol{L}_{22}\boldsymbol{M}^T &= \boldsymbol{0} \end{aligned}$$

thus $\boldsymbol{L}_{22} = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{V}^T\boldsymbol{X}\boldsymbol{U}^T\boldsymbol{\Sigma}^{-\frac{1}{2}}$ and $\boldsymbol{L}$ and its inversion are

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{N} \\ \boldsymbol{N}^T & \boldsymbol{L}_{22} \end{bmatrix}, \ \boldsymbol{L}^{-1} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{M} \\ \boldsymbol{M}^T & \boldsymbol{K} \end{bmatrix}$$

where $\boldsymbol{K}$ is any given matrix of appropriate dimension.

It should be pointed out that there several algorithms exist which give sufficient condition for finding the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ so that $\boldsymbol{Y} = \boldsymbol{X}^1$ via convex optimization (Iwasaki and Skelton, 1995a,b). This immediately allows us to apply LMI methods.

The idea of these algorithms is based on the observation that for any matrices $\boldsymbol{X} \succ 0$ and $\boldsymbol{Y} \succ 0$ that satisfy

$$\begin{bmatrix} \boldsymbol{Y} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{X} \end{bmatrix} \succeq 0 \tag{5.36}$$

the condition trace$(\boldsymbol{XY}) = n$ (where n denotes the number of rows of the matrix $\boldsymbol{X}$ (or $\boldsymbol{Y}$)) implies that $\boldsymbol{XY} = \boldsymbol{I}$. Hence, the solution to the problem of finding the matrices $\boldsymbol{X} \succ 0$ and $\boldsymbol{Y} \succ 0$ such that $\boldsymbol{X} = \boldsymbol{Y}^{-1}$ is reduced to the solution

$$\begin{aligned} &\min_{\boldsymbol{X}\succ 0, \boldsymbol{Y}\succ 0} \quad \mathrm{trace}(\boldsymbol{XY}) \\ &\text{subject to (5.36)} \end{aligned}$$

It was illustrated in (Oliveria and Geromel, 1997) that the above optimization problem can be solved in an effective way using Product Reduction Algorithm (PRA). Moreover, this algorithm is simple in numerical implementation and can therefore be used for our purposes.

5.2.4.1. Product Reduction Algorithm

The preliminary step of this algorithm is to provide any matrices $\boldsymbol{X}_0 \succ 0$ and $\boldsymbol{Y}_0 \succ 0$ which satisfy (5.36) then set the iteration counter $k = 0$ and successively perform the following steps

Step 1: Define the linear function

$$f_k(\boldsymbol{X},\boldsymbol{Y}) = \operatorname{trace}\left(\boldsymbol{X}_k\boldsymbol{Y} + \boldsymbol{Y}_k\boldsymbol{X}\right)$$

Step 2: Compute $\boldsymbol{X}_{k+1}$ and $\boldsymbol{Y}_{k+1}$ by solving linear objective minimization problem of the form

$$\begin{gathered}\min_{\boldsymbol{X}\succ 0,\boldsymbol{Y}\succ 0} \quad f_k(\boldsymbol{X},\boldsymbol{Y}) \\ \text{subject to (5.36)}\end{gathered}$$

Step 3: If prescribed accuracy ϵ is attained e.g. $\operatorname{trace}(\boldsymbol{X}_{k+1}\boldsymbol{Y}+\boldsymbol{Y}_{k+1}\boldsymbol{X})-2n\leq\epsilon$ (where n denotes the number of rows of $\boldsymbol{X}$ or $\boldsymbol{Y}$) then stop, otherwise return to the second step.

5.3. $\mathcal{H}_\infty$ control of uncertain differential LRPs

In this section we extend the results of the previous section to a process model containing some uncertainties in all the matrices. The presence of these uncertainties in the model structure is the consequence of varying physical parameters and imperfect knowledge on process dynamics. Motivated by this practical fact, we are interested in designing a static feedback controller that stabilises the considered class of LRPs and ensures that the $\mathcal{H}_\infty$ norm bound does not exceed the prescribed level for all admissible uncertainties. Two representations of uncertainty are taken into consideration: norm-bounded and polytopic forms. Both of these representations have their advantages and disadvantages and there are also practically relevant problem areas where one in more suitable than the other.

5.3.1. Norm-bounded model of uncertainty

Here, we assume that the uncertainty is norm bounded in both the state and pass profile updating equations. This form corresponds to the process with uncertainty which is modelled as an additive perturbation to the nominal model matrices. In such a case we can write the process state-space model in the form

$$\begin{aligned}\begin{bmatrix}\dot{x}_{k+1}(t)\\ y_{k+1}(t)\end{bmatrix} = {} & \left(\begin{bmatrix}\boldsymbol{A} & \boldsymbol{B}_0\\ \boldsymbol{C} & \boldsymbol{D}_0\end{bmatrix} + \begin{bmatrix}\Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0\\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0\end{bmatrix}\right)\begin{bmatrix}x_{k+1}(t)\\ y_k(t)\end{bmatrix} + \left(\begin{bmatrix}\boldsymbol{B}\\ \boldsymbol{D}\end{bmatrix} + \begin{bmatrix}\Delta\boldsymbol{B}\\ \Delta\boldsymbol{D}\end{bmatrix}\right)u_{k+1}(t)\\ & + \left(\begin{bmatrix}\boldsymbol{B}_1\\ \boldsymbol{D}_1\end{bmatrix} + \begin{bmatrix}\Delta\boldsymbol{B}_1\\ \Delta\boldsymbol{D}_1\end{bmatrix}\right)w_{k+1}(t)\end{aligned} \tag{5.37}$$

where the admissible uncertainties are assumed to be of the form

$$\begin{bmatrix}\Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0 & \Delta\boldsymbol{B} & \Delta\boldsymbol{B}_1\\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 & \Delta\boldsymbol{D} & \Delta\boldsymbol{D}_1\end{bmatrix} = \begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\end{bmatrix}\boldsymbol{\mathcal{F}}\begin{bmatrix}\boldsymbol{E}_1 & \boldsymbol{E}_2 & \boldsymbol{E}_3 & \boldsymbol{E}_4\end{bmatrix} \tag{5.38}$$

and where $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$, $\boldsymbol{E}_4$ are known constant matrices of compatible dimensions, and $\boldsymbol{\mathcal{F}}$ is an unknown matrix with constant entries which satisfies

(4.5). In this case, the following theorem gives the condition of a controller existing that stabilises the process (5.37) and ensures that the $\mathcal{H}_\infty$ norm bound does not exceed prescribed level for all admissible uncertainties.

Theorem 5.5. *Suppose that a differential LRP of the form described by (5.37), with uncertainty structure modelled by (5.38) and (4.5), is subjected to a control law of the form (4.17). Then the resulting closed-loop process is stable along the pass for all admissible uncertainties and has the prescribed $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$ if there exist matrices $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ of compatible dimensions and a scalar $\epsilon > 0$ such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{W}_2+3\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1+3\epsilon\boldsymbol{H}_2\boldsymbol{H}_1^T & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{D}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T+3\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1+3\epsilon\boldsymbol{H}_1\boldsymbol{H}_1^T & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{B}_1 & \boldsymbol{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T & \boldsymbol{0} \\ \boldsymbol{D}_1^T & \boldsymbol{B}_1^T & \boldsymbol{0} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_4^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_4 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.39}$$

If (5.39) holds then the controller matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are given by (4.20).

Proof. *In view of the proof of Theorem 5.2 it can be shown that the closed-loop process is stable along the pass for all admissible uncertainties and has prescribed $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$ if the following inequality is satisfied*

$$\begin{bmatrix} \boldsymbol{0} & \Delta\boldsymbol{C}\boldsymbol{W}_1+\Delta\boldsymbol{D}\boldsymbol{N}_1 & \Delta\boldsymbol{D}_0\boldsymbol{W}_2+\Delta\boldsymbol{D}\boldsymbol{N}_2 & \Delta\boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{W}_1\Delta\boldsymbol{C}^T+\boldsymbol{N}_1^T\Delta\boldsymbol{D}^T & \Delta\boldsymbol{\Lambda} & \Delta\boldsymbol{B}_0\boldsymbol{W}_2+\Delta\boldsymbol{B}\boldsymbol{N}_2 & \Delta\boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{W}_2\Delta\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\Delta\boldsymbol{D}^T & \boldsymbol{W}_2\Delta\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\Delta\boldsymbol{B}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \Delta\boldsymbol{D}_1^T & \Delta\boldsymbol{B}_1^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & \boldsymbol{\Lambda} & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2 \\ \boldsymbol{D}_1^T & \boldsymbol{B}_1^T & \boldsymbol{0} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.40}$$

where $\boldsymbol{\Lambda} = \boldsymbol{W}_1\boldsymbol{A}^T + \boldsymbol{N}_1^T\boldsymbol{B}^T + \boldsymbol{A}\boldsymbol{W}_1 + \boldsymbol{B}\boldsymbol{N}_1$ and $\Delta\boldsymbol{\Lambda} = \boldsymbol{W}_1\Delta\boldsymbol{A}^T + \boldsymbol{N}_1^T\Delta\boldsymbol{B}^T + \Delta\boldsymbol{A}\boldsymbol{W}_1 + \Delta\boldsymbol{B}\boldsymbol{N}_1$. The first term in the above inequality can be rewritten as

$$\overline{\boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}} + \overline{\boldsymbol{E}}^T\overline{\boldsymbol{\mathcal{F}}}^T\overline{\boldsymbol{H}}^T$$

where

$$\overline{\boldsymbol{H}} = \begin{bmatrix} \mathbf{0} & \boldsymbol{H}_2 & \boldsymbol{H}_2 & \boldsymbol{H}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_1 & \boldsymbol{H}_1 & \boldsymbol{H}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \; \overline{\boldsymbol{\mathcal{F}}} = \mathrm{diag}\left(\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}\right),$$

$$\overline{\boldsymbol{E}} = \mathrm{diag}\left(\mathbf{0}, \boldsymbol{E}_1\boldsymbol{W}_1 + \boldsymbol{E}_3\boldsymbol{N}_1, \boldsymbol{E}_2\boldsymbol{W}_2 + \boldsymbol{E}_3\boldsymbol{N}_2, \boldsymbol{E}_4, \mathbf{0}\right)$$

An obvious application of Lemma 8 followed by application of the Schur complement formula yields (5.39) and the proof is complete. ■

5.3.2. Polytopic model of uncertainty

Consider the differential LRP which the current pass state updating equation matrices range in the polytope of matrices as it is described by

$$\left[\, \boldsymbol{A} \quad \boldsymbol{B} \quad \boldsymbol{B}_0 \quad \boldsymbol{B}_1 \,\right] \in \mathrm{Co}\left(\left[\, \boldsymbol{A}^i \quad \boldsymbol{B}_0^i \quad \boldsymbol{B}^i \quad \boldsymbol{B}_1 \,\right]\right), \; i = 1, 2, \ldots, h \tag{5.41}$$

and

$$\mathrm{Co}\left(\left[\, \boldsymbol{A}^i \; \boldsymbol{B}_0^i \; \boldsymbol{B}^i \; \boldsymbol{B}_1^i \,\right]\right) := \left\{ \boldsymbol{X} : \boldsymbol{X} = \sum_{i=1}^{h} \alpha_i \left[\, \boldsymbol{A}^i \; \boldsymbol{B}_0^i \; \boldsymbol{B}^i \; \boldsymbol{B}_1^i \,\right], \alpha_i \geq 0, \; \sum_{i=1}^{h} \alpha_i = 1 \right\} \tag{5.42}$$

For the current pass profile updating equation in (5.2) we assume a norm-bounded type of uncertainty

$$\begin{aligned} y_{k+1}(t) =& (\boldsymbol{C} + \Delta\boldsymbol{C}) x_{k+1}(t) + (\boldsymbol{D}_0 + \Delta\boldsymbol{D}_0) y_k(t) \\ &+ (\boldsymbol{D} + \Delta\boldsymbol{D}) u_{k+1}(t) + (\boldsymbol{D}_1 + \Delta\boldsymbol{D}_1) w_{k+1}(t) \end{aligned}$$

where

$$\left[\; \Delta\boldsymbol{C} \; \Delta\boldsymbol{D}_0 \; \Delta\boldsymbol{D} \; \Delta\boldsymbol{D}_1 \;\right] = \boldsymbol{H}_2 \boldsymbol{\mathcal{F}} \left[\; \boldsymbol{E}_1 \; \boldsymbol{E}_2 \; \boldsymbol{E}_3 \; \boldsymbol{E}_4 \;\right] \tag{5.43}$$

and the matrix $\boldsymbol{\mathcal{F}}$ satisfies (4.5).

Theorem 5.6. *Suppose that a differential LRP of the form described by (5.2), with uncertainty structure modelled by (5.41)-(5.42) and (5.43), is subjected to a control law of the form of (4.17). Then the resulting closed-loop process is stable along the pass for all admissible uncertainties and has the prescribed $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$ if there exist matrices $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ of* *compatible dimensions and a scalar $\epsilon > 0$, such that*

$$\begin{bmatrix} -\boldsymbol{W}_2+3\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{D}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & \boldsymbol{W}_1\boldsymbol{A}^{iT}+\boldsymbol{N}_1^T\boldsymbol{B}^{iT}+\boldsymbol{A}^i\boldsymbol{W}_1+\boldsymbol{B}^i\boldsymbol{N}_1 & \boldsymbol{B}_0^i\boldsymbol{W}_2+\boldsymbol{B}^i\boldsymbol{N}_2 & \boldsymbol{B}_1^i & \boldsymbol{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^{iT}+\boldsymbol{N}_2^T\boldsymbol{B}^{iT} & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T & \boldsymbol{0} \\ \boldsymbol{D}_1^T & \boldsymbol{B}_1^{iT} & \boldsymbol{0} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_4^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_4 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.44}$$

If the LMI (5.44) holds then the controller matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are given by (4.20).

Proof. *It can be proven in an identical manner to proofs of previously presented Theorems.* ■

5.4. $\mathcal{H}_\infty$ control of discrete LRPs

This section addresses the problem of the $\mathcal{H}_\infty$ disturbance rejection or attenuation problem for discrete LRP. Therefore, the aim can be summarized as finding an implementable control law which will give stability along the pass closed loop with a prescribed degree of disturbance rejection, including the case when there is an uncertainty in the model structure.

5.4.1. LMI-based $\mathcal{H}_\infty$ norm computation

Consider the model of a discrete LRP (5.3) with no control inputs but with external disturbance inputs given by

$$\begin{bmatrix} x_{k+1}(p+1) \\ y_{k+1}(p) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_0 \\ \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix} \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{D}_1 \end{bmatrix} w_{k+1}(p) \tag{5.45}$$

then with (5.4) we have the following result.

Theorem 5.7. *A discrete LRP described by (5.45) is stable along the pass and has the $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$ if there exist matrices $\boldsymbol{P}_1 \succ 0$ and $\boldsymbol{P}_2 \succ 0$ such that the following LMI with $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{P}_1, \boldsymbol{P}_2)$ holds*

$$\begin{bmatrix} -\boldsymbol{P} & \boldsymbol{P}\boldsymbol{\Phi} & \boldsymbol{P}\boldsymbol{\Omega} & \boldsymbol{0} \\ \boldsymbol{\Phi}^T\boldsymbol{P} & -\boldsymbol{P} & \boldsymbol{0} & \boldsymbol{C}_2^T \\ \boldsymbol{\Omega}^T\boldsymbol{P} & \boldsymbol{0} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_2 & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.46}$$

where $\boldsymbol{\Phi}$, $\boldsymbol{\Omega}$ and $\boldsymbol{C}_2$ are defined in (5.16).

Proof. *In order to ensure the $\mathcal{H}_\infty$ noise attenuation γ holds, it is required that the associated Hamiltonian defined by*

$$H(k,p) = \Delta V(k,p) + y_{k+1}^T(p) y_{k+1}(p) - \gamma^2 w_{k+1}^T(p) w_{k+1}(p) \tag{5.47}$$

satisfies

$$H(k,p) < 0 \tag{5.48}$$

where $\Delta V(k,p)$ is defined in (4.34). Next, by carrying out appropriate substitutions we obtain

$$H(k,p) = \begin{bmatrix} \zeta^T(k,p) & w_{k+1}^T(p) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}^T \boldsymbol{P} \boldsymbol{\Phi} - \boldsymbol{P} + \boldsymbol{C}_2^T \boldsymbol{C}_2 & \boldsymbol{\Phi}^T \boldsymbol{P} \boldsymbol{\Omega} \\ \boldsymbol{\Omega}^T \boldsymbol{P} \boldsymbol{\Phi} & \boldsymbol{\Omega} \boldsymbol{P} \boldsymbol{\Omega} - \gamma^2 \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \zeta(k,p) \\ w_{k+1}(p) \end{bmatrix}$$

where $\zeta(k,p)$ is defined in (4.32). Now the above condition guarantees that (5.48) holds for any $\zeta(k,p), w_{k+1}(p) \neq 0$. Finally, application of the Schur complement formula gives the solution and the proof is complete. ■

Remark 5.2. *Since LRPs share certain structural similarities with 2-D linear systems then the result can be also obtained via manipulations required for 2-D linear systems described by RM, for example, in (Du and Xie, 2002). On the other hand, it should be pointed out that the structure of the control algorithms are not well founded physically due to the fact that, for example, the concept of a state for these systems is not uniquely defined. Furthermore, it turns out that, in case of LRP, a control law is a combination of current pass information and 'feedforward' information from the previous pass. In terms of 2-D systems this is the equivalent to a clear state feedback control law, which has not been used in (Du and Xie, 2002) for $\mathcal{H}_\infty$ control. Moreover, it is known in terms of RM, that the pass profile vector is simultaneously the output vector which employs different algebraic manipulations and in effect yields a simplified version in comparison to the standard 2-D approach of (Du and Xie, 2002; Du et al., 2001).*

Remark 5.3. *It can be seen that the inequality (5.46) has the same form as the inequality (4.46). Therefore, the robust stabilisation of an uncertain LRP can be solved via $\mathcal{H}_\infty$ control results developed here.*

5.4.2. $\mathcal{H}_\infty$ control with a static feedback controller

The following result enables (4.40) to be designed to give stability along the pass of the closed-loop process with a prescribed disturbance rejection bound.

Theorem 5.8. *Suppose that a discrete LRP described by (5.3) is subject to a control law of the form (4.40). Then the resulting closed-loop process is stable along the pass with the prescribed $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$ if there exists* *matrices $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{W}_1 & \boldsymbol{0} & \boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T & \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & -\boldsymbol{W}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{0} & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2 \\ \boldsymbol{B}_1^T & \boldsymbol{D}_1^T & \boldsymbol{0} & \boldsymbol{0} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.49}$$

and the required controller matrices in (4.40) are given by (4.20).

Proof. *Applying the result of Theorem 5.7 it follows immediately that stability along the pass holds if there exist matrices $\boldsymbol{P}_1 \succ 0$ and $\boldsymbol{P}_2 \succ 0$ such that*

$$\begin{bmatrix} -\boldsymbol{P}_1 & \boldsymbol{0} & \boldsymbol{P}_1\boldsymbol{A}+\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{\Lambda}_1 & \boldsymbol{P}_1\boldsymbol{B}_1 & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C}+\boldsymbol{P}_2\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{\Lambda}_2 & \boldsymbol{P}_2\boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{K}_1^T\boldsymbol{B}^T\boldsymbol{P}_1 & \boldsymbol{C}^T\boldsymbol{P}_2+\boldsymbol{K}_1^T\boldsymbol{D}^T\boldsymbol{P}_2 & -\boldsymbol{P}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Lambda}_1^T & \boldsymbol{\Lambda}_2^T & \boldsymbol{0} & -\boldsymbol{P}_2 & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{B}_1^T\boldsymbol{P}_1 & \boldsymbol{D}_1^T\boldsymbol{P}_2 & \boldsymbol{0} & \boldsymbol{0} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0$$

where $\boldsymbol{\Lambda}_1 = \boldsymbol{P}_1\boldsymbol{B}_0+\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{K}_2$ and $\boldsymbol{\Lambda}_2 = \boldsymbol{P}_2\boldsymbol{D}_0+\boldsymbol{P}_2\boldsymbol{D}\boldsymbol{K}_2$, This last inequality is not in LMI form because it is nonlinear with respect to its parameters. Consequently, set $\boldsymbol{P}_1 = \boldsymbol{W}_1^{-1}$, $\boldsymbol{P}_2 = \boldsymbol{W}_2^{-1}$ and pre and post-multiply by $\operatorname{diag}(\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{I}, \boldsymbol{I})$*, followed by setting $\boldsymbol{N}_1 = \boldsymbol{K}_1\boldsymbol{W}_1$ and $\boldsymbol{N}_2 = \boldsymbol{K}_2\boldsymbol{W}_2$ to obtain the LMI of (5.49) and the proof is complete.* ■

Note that the $\mathcal{H}_\infty$ disturbance attenuation here can be minimized using the linear objective minimization procedure which leads, in effect, to minimization of the effects of the disturbance.

$$\min_{\boldsymbol{W}_1\succ 0, \boldsymbol{W}_2\succ 0, \boldsymbol{N}_1, \boldsymbol{N}_2} \quad \mu$$
$$\text{subject to (5.49) with } \mu = \gamma^2$$

5.4.3. $\mathcal{H}_\infty$ control with a dynamic pass profile controller

The control law of the previous section requires that the complete current pass state vector is available for measurement. If this is not the case then one option is to use an observer to reconstruct it. In this section, we consider the control of processes described by (5.3) through use of a full dynamic pass profile feedback controller of the order n_c

$$\begin{aligned} \begin{bmatrix} x_{k+1}^c(p+1) \\ y_{k+1}^c(p) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}_{c11} & \boldsymbol{A}_{c12} \\ \boldsymbol{A}_{c21} & \boldsymbol{A}_{c22} \end{bmatrix} \begin{bmatrix} x_{k+1}^c(p) \\ y_k^c(p) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_{c1} \\ \boldsymbol{B}_{c2} \end{bmatrix} y_{k+1}(p) \\ u_{k+1}(p) &= \begin{bmatrix} \boldsymbol{C}_{c1} & \boldsymbol{C}_{c2} \end{bmatrix} \begin{bmatrix} x_{k+1}^c(p) \\ y_k^c(p) \end{bmatrix} + \boldsymbol{D}_c y_{k+1}(p) \end{aligned} \tag{5.50}$$

where $x_k^c(p) \in \mathbb{R}^{n_1}$ is the controller state vector, $y_k^c(p) \in \mathbb{R}^{m_1}$ is the output vector and $n_c = n_1 + m_1$.

To obtain the state-space model of the resulting closed-loop process, use the notation (5.16)-(5.17) and introduce the so-called augmented state and pass profile vectors for the closed-loop process as

$$\overline{x}_{k+1}(p) = \begin{bmatrix} x_{k+1}(p) \\ x^c_{k+1}(p) \end{bmatrix}, \ \overline{y}_k(p) = \begin{bmatrix} y_k(p) \\ y^c_k(p) \end{bmatrix}$$

Then the closed-loop process is given by

$$\begin{aligned} \begin{bmatrix} \overline{x}_{k+1}(p+1) \\ \overline{y}_{k+1}(p) \end{bmatrix} &= \overline{\boldsymbol{A}} \begin{bmatrix} \overline{x}_{k+1}(p) \\ \overline{y}_k(p) \end{bmatrix} + \overline{\boldsymbol{B}} w_k(p) \\ y_{k+1}(p) &= \overline{\boldsymbol{C}} \begin{bmatrix} \overline{x}_{k+1}(p) \\ \overline{y}_k(p) \end{bmatrix} \end{aligned} \tag{5.51}$$

where

$$\overline{\boldsymbol{A}} = \boldsymbol{\Pi} \begin{bmatrix} \boldsymbol{\Phi} + \boldsymbol{B}_2 \boldsymbol{D}_c \boldsymbol{C}_2 & \boldsymbol{B}_2 \boldsymbol{C}_c \\ \boldsymbol{B}_c \boldsymbol{C}_2 & \boldsymbol{A}_c \end{bmatrix} \boldsymbol{\Pi}^T, \ \overline{\boldsymbol{B}} = \boldsymbol{\Pi} \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{0} \end{bmatrix}, \ \overline{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C}_2 & \boldsymbol{0} \end{bmatrix} \boldsymbol{\Pi}^T$$

Next, define

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \ \boldsymbol{\mathcal{B}}_2 = \begin{bmatrix} \boldsymbol{B}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \ \boldsymbol{\mathcal{C}}_2 = \begin{bmatrix} \boldsymbol{C}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \ \boldsymbol{\mathcal{C}} = \begin{bmatrix} \boldsymbol{C}_2 & \boldsymbol{0} \end{bmatrix}, \ \boldsymbol{\mathcal{B}} = \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{0} \end{bmatrix}$$

then it is obvious that the closed-loop state-space model matrices can be written in the following form which is affine in the controller data matrix $\boldsymbol{\Theta}$ (introduced in (5.19))

$$\overline{\boldsymbol{A}} = \boldsymbol{\Pi} \left[\ \boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}}_2 \boldsymbol{\Theta} \boldsymbol{\mathcal{C}}_2 \ \right] \boldsymbol{\Pi}^T, \ \overline{\boldsymbol{C}} = \boldsymbol{\mathcal{C}} \boldsymbol{\Pi}^T, \ \overline{\boldsymbol{B}} = \boldsymbol{\Pi} \boldsymbol{\mathcal{B}} \tag{5.52}$$

Now we have the following result which gives an existence condition for the controller matrices $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$, $\boldsymbol{D}_c$ to ensure stability along the pass and then enables controller design.

Theorem 5.9. *Suppose that a dynamic pass profile feedback controller defined by (5.50) is applied to a discrete LRP described by (5.3) with resulting closed-loop state-space model (5.51). Suppose also that there exist matrices* $\boldsymbol{P}_{11} \succ 0$, *(*$\boldsymbol{P}_{11} = \mathrm{diag}(\boldsymbol{P}_{h11}, \boldsymbol{P}_{v11})$*) and* $\boldsymbol{R}_{11} \succ 0$, *(*$\boldsymbol{R}_{11} = \mathrm{diag}(\boldsymbol{R}_{h11}, \boldsymbol{R}_{v11})$*) such that the LMIs defined by (5.53)–(5.55) below hold. Then the resulting closed-loop process is stable along the pass and has the* $\mathcal{H}_\infty$ *disturbance attenuation* $\gamma > 0$

$$\begin{bmatrix} \boldsymbol{\mathcal{N}}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Phi} \boldsymbol{R}_{11} \boldsymbol{\Phi}^T - \boldsymbol{R}_{11} & \boldsymbol{\Omega} & \boldsymbol{\Phi} \boldsymbol{R}_{11} \boldsymbol{C}_2^T \\ \boldsymbol{\Omega}^T & -\gamma^2 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{C}_2 \boldsymbol{R}_{11} \boldsymbol{\Phi}^T & \boldsymbol{0} & -\boldsymbol{I} + \boldsymbol{C}_2 \boldsymbol{R}_{11} \boldsymbol{C}_2^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{N}}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.53}$$

$$\begin{bmatrix} \boldsymbol{\mathcal{N}}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Phi}^T \boldsymbol{P}_{11} \boldsymbol{\Phi} - \boldsymbol{P}_{11} & \boldsymbol{\Phi}^T \boldsymbol{P}_{11} \boldsymbol{\Omega} & \boldsymbol{C}_2^T \\ \boldsymbol{\Omega} \boldsymbol{P}_{11} \boldsymbol{\Phi} & \boldsymbol{\Omega}^T \boldsymbol{P}_{11} \boldsymbol{\Omega} - \gamma^2 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{C}_2 & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{N}}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.54}$$

$$\begin{bmatrix} \boldsymbol{P}_{h11} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{R}_{h11} \end{bmatrix} \succeq 0, \ \begin{bmatrix} \boldsymbol{P}_{v11} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{R}_{v11} \end{bmatrix} \succeq 0 \tag{5.55}$$

where $\boldsymbol{\mathcal{N}}_1$ and $\boldsymbol{\mathcal{N}}_2$ are full column rank matrices whose images satisfy

$$\mathrm{Im}\,(\boldsymbol{\mathcal{N}}_1) = \ker\left(\boldsymbol{B}_2^T\right),\ \mathrm{Im}\,(\boldsymbol{\mathcal{N}}_2) = \ker(\boldsymbol{C}_2)$$

Proof. *Interpreting the result of Theorem 5.7 in terms of the matrices (5.52) gives the following inequality*

$$\boldsymbol{\Psi} + \boldsymbol{M}^T\boldsymbol{\Theta}\boldsymbol{N} + \boldsymbol{N}^T\boldsymbol{\Theta}^T\boldsymbol{M} \prec 0 \tag{5.56}$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} -\boldsymbol{R} & \boldsymbol{R}\boldsymbol{\mathcal{A}} & \boldsymbol{R}\boldsymbol{\mathcal{B}} & \boldsymbol{0} \\ \boldsymbol{\mathcal{A}}^T\boldsymbol{R} & -\boldsymbol{R} & \boldsymbol{0} & \boldsymbol{\mathcal{C}}^T \\ \boldsymbol{\mathcal{B}}^T\boldsymbol{R} & \boldsymbol{0} & -\gamma^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{C}} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix},\ \boldsymbol{M}^T = \begin{bmatrix} \boldsymbol{R}\boldsymbol{\mathcal{B}}_2 \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix},\ \boldsymbol{N} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\mathcal{C}}_2 & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

and $\boldsymbol{R} = \boldsymbol{\Pi}^T\boldsymbol{P}\boldsymbol{\Pi}$. Partitioning the matrix $\boldsymbol{R}$ as

$$\boldsymbol{R} = \boldsymbol{\Pi}^T\boldsymbol{P}\boldsymbol{\Pi} = \left[\begin{array}{cc|cc} \boldsymbol{P}_{h_{11}} & \boldsymbol{0} & \boldsymbol{P}_{h_{12}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_{v_{11}} & \boldsymbol{0} & \boldsymbol{P}_{v_{12}} \\ \hline \boldsymbol{P}_{h_{12}}^T & \boldsymbol{0} & \boldsymbol{P}_{h_{22}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_{v_{12}}^T & \boldsymbol{0} & \boldsymbol{P}_{v_{22}} \end{array}\right] = \left[\begin{array}{c|c} \boldsymbol{P}_{11} & \boldsymbol{P}_{12} \\ \hline \boldsymbol{P}_{12}^T & \boldsymbol{P}_{22} \end{array}\right] \tag{5.57}$$

where

$$\boldsymbol{P}_h = \begin{bmatrix} \boldsymbol{P}_{h_{11}} & \boldsymbol{P}_{h_{12}} \\ \boldsymbol{P}_{h_{12}}^T & \boldsymbol{P}_{h_{22}} \end{bmatrix},\ \boldsymbol{P}_v = \begin{bmatrix} \boldsymbol{P}_{v_{11}} & \boldsymbol{P}_{v_{12}} \\ \boldsymbol{P}_{v_{12}}^T & \boldsymbol{P}_{v_{22}} \end{bmatrix}$$

it can be written that

$$\boldsymbol{P}_h^{-1} = \begin{bmatrix} \boldsymbol{R}_{h_{11}} & \boldsymbol{R}_{h_{12}} \\ \boldsymbol{R}_{h_{12}}^T & \boldsymbol{R}_{h_{22}} \end{bmatrix},\ \boldsymbol{P}_v^{-1} = \begin{bmatrix} \boldsymbol{R}_{v_{11}} & \boldsymbol{R}_{v_{12}} \\ \boldsymbol{R}_{v_{12}}^T & \boldsymbol{R}_{v_{22}} \end{bmatrix}$$

and

$$\boldsymbol{R}^{-1} = \boldsymbol{\Pi}^T\boldsymbol{P}^{-1}\boldsymbol{\Pi} = \left[\begin{array}{cc|cc} \boldsymbol{R}_{h_{11}} & \boldsymbol{0} & \boldsymbol{R}_{h_{12}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_{v_{11}} & \boldsymbol{0} & \boldsymbol{R}_{v_{12}} \\ \hline \boldsymbol{R}_{h_{12}}^T & \boldsymbol{0} & \boldsymbol{R}_{h_{22}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_{v_{12}}^T & \boldsymbol{0} & \boldsymbol{R}_{v_{22}} \end{array}\right] = \left[\begin{array}{c|c} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \hline \boldsymbol{R}_{12}^T & \boldsymbol{R}_{22} \end{array}\right]$$

After manipulations similar to these described in the proof of Theorem 5.4 the conditions of (5.53) and (5.54) are obtained.

The last problem is to provide the conditions which allow us to find the matrix $\boldsymbol{P}$ and its inverse. To begin, first note again that $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{P}_h, \boldsymbol{P}_v)$ and that only $\boldsymbol{P}_{11}$ and $\boldsymbol{R}_{11}$ appear in the first two LMIs to be satisfied. Application of Lemma 10 now gives the required conditions and the proof is complete. ■

If this last result holds then the stabilising controller can be designed using the following algorithm

Step 1: Compute the matrices $\boldsymbol{P}_{h12}$, $\boldsymbol{P}_{v12}$ using

$$\begin{aligned}\boldsymbol{P}_{h12}\boldsymbol{P}_{h22}^{-1}\boldsymbol{P}_{h12}^{T} &= \boldsymbol{P}_{h11} - \boldsymbol{R}_{h11}^{-1},\\ \boldsymbol{P}_{v12}\boldsymbol{P}_{v22}^{-1}\boldsymbol{P}_{v12}^{T} &= \boldsymbol{P}_{v11} - \boldsymbol{R}_{v11}^{-1}\end{aligned}$$

where $\boldsymbol{P}_{h22} = \boldsymbol{I}$ and $\boldsymbol{P}_{v22} = \boldsymbol{I}$

Step 2: Construct $\boldsymbol{P}_h \succ 0$ and $\boldsymbol{P}_v \succ 0$ and then the matrix $\boldsymbol{P} = \text{diag}(\boldsymbol{P}_h, \boldsymbol{P}_v)$.

Step 3: Compute the matrices $\boldsymbol{M}$, $\boldsymbol{N}$ and $\boldsymbol{\Psi}$.

Step 4: Solve the LMI (5.56) (where $\boldsymbol{\Theta}$ is the unknown matrix) and hence the controller state-space model matrices.

Also the disturbance attenuation level γ can be minimized using the following optimization procedure

$$\min_{\boldsymbol{P}_{11}\succ 0,\ \boldsymbol{R}_{11}\succ 0} \quad \mu$$
$$\text{subject to } (5.53) - (5.55) \text{ with } \mu = \gamma^2$$

5.5. $\mathcal{H}_\infty$ control of uncertain discrete LRPs

This section deals with the robust $\mathcal{H}_\infty$ control problem for discrete LRPs with norm-bounded uncertainty. For this purpose consider a discrete LRP with uncertainty modelled as additive perturbations to the nominal model matrices with resulting state-space model

$$\begin{aligned}x_{k+1}(p+1) =&(\boldsymbol{A}+\Delta\boldsymbol{A})x_{k+1}(p) + (\boldsymbol{B}+\Delta\boldsymbol{B})u_{k+1}(p) + (\boldsymbol{B}_0+\Delta\boldsymbol{B}_0)y_k(p)\\ &+ (\boldsymbol{B}_1+\Delta\boldsymbol{B}_1)w_{k+1}(p)\\ y_{k+1}(p) =&(\boldsymbol{C}+\Delta\boldsymbol{C})x_{k+1}(p) + (\boldsymbol{D}+\Delta\boldsymbol{D})u_{k+1}(p) + (\boldsymbol{D}_0+\Delta\boldsymbol{D}_0)y_k(p)\\ &+ (\boldsymbol{D}_1+\Delta\boldsymbol{D}_1)w_{k+1}(p)\end{aligned} \tag{5.58}$$

where the admissible uncertainties to be of the form (5.38) and (4.5). Now we have the following result.

Theorem 5.10. *Suppose that a discrete LRP described by (5.58) with the uncertainty structure satisfying (5.38) and (4.5) is subject to a control law defined by (4.40). Then the resulting closed-loop process is stable along the pass with the prescribed $\mathcal{H}_\infty$ disturbance attenuation $\gamma > 0$ if there exist a scalar $\epsilon > 0$ and* *matrices $\boldsymbol{W}_1 \succ 0$, $\boldsymbol{W}_2 \succ 0$, and $\boldsymbol{N}_1$, $\boldsymbol{N}_2$ such that the following LMI holds*

$$\left[\begin{array}{cccc} -\boldsymbol{W}_1 + 3\epsilon\boldsymbol{H}_1\boldsymbol{H}_1^T & 3\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & \boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 \\ 3\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & -\boldsymbol{W}_2 + 3\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2^T \\ \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T & \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & -\boldsymbol{W}_1 & \mathbf{0} \\ \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \mathbf{0} & -\boldsymbol{W}_2 \\ \boldsymbol{B}_1^T & \boldsymbol{D}_1^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{W}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right.$$

$$\left.\begin{array}{ccccc} \boldsymbol{B}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{D}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{W}_2 & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \mathbf{0} & \mathbf{0} \\ -\gamma^2\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_4 \\ \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T & \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{E}_4^T & \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} \end{array}\right] \prec 0 \tag{5.59}$$

If this condition holds, the corresponding controller matrices are given by (4.20).

Proof. *On applying (4.40), the closed-loop process stability along the pass condition can be written in the form*

$$\boldsymbol{\Gamma} + \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{\mathcal{F}}}\widetilde{\boldsymbol{E}} + \widetilde{\boldsymbol{E}}^T\widetilde{\boldsymbol{\mathcal{F}}}^T\widetilde{\boldsymbol{H}}^T \prec 0$$

where $\boldsymbol{\Gamma}$ is left hand side of LMI (5.49) and

$$\widetilde{\boldsymbol{H}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{H}_1 & \boldsymbol{H}_1 & \boldsymbol{H}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{H}_2 & \boldsymbol{H}_2 & \boldsymbol{H}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \begin{array}{l} \widetilde{\boldsymbol{F}} = \operatorname{diag}(\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}), \\ \widetilde{\boldsymbol{E}} = \operatorname{diag}(\mathbf{0}, \mathbf{0}, \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1, \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2, \boldsymbol{E}_4, \mathbf{0}) \end{array}$$

The LMI of (5.59) is now obtained by an application of the inequality of Lemma 8 followed by the Schur complement formula and the proof is complete. ■

To reduce the effects of the disturbance vector, the following linear objective minimization procedure can be used.

$$\begin{gathered} \min_{\boldsymbol{W}_1 \succ 0, \boldsymbol{W}_2 \succ 0, \boldsymbol{N}_1, \boldsymbol{N}_2, \epsilon > 0} \mu \\ \text{subject to (5.59) with } \mu = \gamma^2 \end{gathered}$$

5.5.1. Alternative robust stabilisation

In this subsection the solution for the $\mathcal{H}_\infty$ problem is applied to design a pass profile feedback controller in the presence of norm-bounded parameter uncertainties.

Consider the state-space model of a discrete LRP written in the form (4.29). Further, use the notation (5.16)-(5.17) and introduce

$$\Delta\boldsymbol{\Phi} = \begin{bmatrix}\Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0\\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0\end{bmatrix} = \begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\end{bmatrix}\gamma^{-1}\boldsymbol{\mathcal{F}}\begin{bmatrix}\boldsymbol{E}_1 & \boldsymbol{E}_2\end{bmatrix},$$
$$\Delta\boldsymbol{B}_2 = \begin{bmatrix}\Delta\boldsymbol{B}\\ \Delta\boldsymbol{D}\end{bmatrix} = \begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\end{bmatrix}\gamma^{-1}\boldsymbol{\mathcal{F}}\begin{bmatrix}\boldsymbol{E}_3\end{bmatrix}$$

The matrices $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ are known and constant and a scalar $\gamma > 0$ is given, hence they are defined in the same form as in (4.30) and the matrix $\boldsymbol{\mathcal{F}}$ satisfies (4.5). In the case when the pass profile controller (5.50) is applied, the closed loop process state space model can be written as

$$\begin{aligned}\begin{bmatrix}\overline{x}_{k+1}(p+1)\\ \overline{y}_k(p+1)\end{bmatrix} &= (\overline{\boldsymbol{A}} + \Delta\overline{\boldsymbol{A}})\begin{bmatrix}\overline{x}_{k+1}(p)\\ \overline{y}_k(p)\end{bmatrix}\\ y_{k+1}(p) &= \overline{\boldsymbol{C}}\begin{bmatrix}\overline{x}_{k+1}(p)\\ \overline{y}_k(p)\end{bmatrix}\end{aligned} \tag{5.60}$$

with

$$\begin{aligned}\overline{\boldsymbol{A}}+\Delta\overline{\boldsymbol{A}} &= \boldsymbol{\Pi}\begin{bmatrix}\boldsymbol{\Phi}+\boldsymbol{B}_2\boldsymbol{D}_c\boldsymbol{C}_2 & \boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{B}_c\boldsymbol{C}_2 & \boldsymbol{A}_c\end{bmatrix}\boldsymbol{\Pi}^T+\boldsymbol{\Pi}\begin{bmatrix}\Delta\boldsymbol{\Phi}+\Delta\boldsymbol{B}_2\boldsymbol{D}_c\boldsymbol{C}_2 & \Delta\boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Pi}^T\\ &= \boldsymbol{\Pi}\begin{bmatrix}\boldsymbol{\Phi}+\boldsymbol{B}_2\boldsymbol{D}_c\boldsymbol{C}_2 & \boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{B}_c\boldsymbol{C}_2 & \boldsymbol{A}_c\end{bmatrix}\boldsymbol{\Pi}^T+\boldsymbol{\Pi}\begin{bmatrix}\gamma^{-1}\boldsymbol{H}\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{\mathcal{F}}\begin{bmatrix}\boldsymbol{E}+\boldsymbol{E}_3\boldsymbol{D}_c\boldsymbol{C}_2 & \boldsymbol{E}_3\boldsymbol{C}_c\end{bmatrix}\boldsymbol{\Pi}^T\\ &= \overline{\boldsymbol{A}} + \overline{\boldsymbol{H}}\boldsymbol{\mathcal{F}}\overline{\boldsymbol{E}}\end{aligned}$$

where the matrices $\boldsymbol{\Phi}$, $\boldsymbol{B}_2$, $\boldsymbol{C}_2$ are as before and

$$\boldsymbol{H} = \begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\end{bmatrix},\ \boldsymbol{E} = \begin{bmatrix}\boldsymbol{E}_1 & \boldsymbol{E}_2\end{bmatrix} \tag{5.61}$$

Now we have the following result.

Theorem 5.11. *Consider a discrete LRP whose dynamics are described by (5.58). Suppose also that a full dynamic pass profile feedback controller defined by (5.50) is applied. Then the resulting closed-loop process (5.60) is stable along the pass holds if there exist matrices $\boldsymbol{P}_{11} \succ 0$, ($\boldsymbol{P}_{11} = \operatorname{diag}(\boldsymbol{P}_{h11}, \boldsymbol{P}_{v11})$) and $\boldsymbol{R}_{11} \succ 0$, ($\boldsymbol{R}_{11} = \operatorname{diag}(\boldsymbol{R}_{h11}, \boldsymbol{R}_{v11})$) such that the following LMIs hold*

$$\begin{bmatrix}\boldsymbol{\mathcal{N}}_1 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{\Phi}^T\boldsymbol{P}_{11}\boldsymbol{\Xi}-\boldsymbol{P}_{11} & \boldsymbol{\Phi}^T\boldsymbol{P}_{11}\boldsymbol{H} & \boldsymbol{E}^T\\ \boldsymbol{H}^T\boldsymbol{P}_{11}\boldsymbol{\Phi} & \boldsymbol{H}^T\boldsymbol{P}_{11}\boldsymbol{H}-\gamma^2\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{E} & \boldsymbol{0} & -\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\mathcal{N}}_1 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}\end{bmatrix} \prec 0 \tag{5.62}$$

$$\begin{bmatrix} \boldsymbol{\mathcal{N}}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Phi}\boldsymbol{R}_{11}\boldsymbol{\Phi}^T - \boldsymbol{R}_{11} & \boldsymbol{\Phi}\boldsymbol{R}_{11}\boldsymbol{E}^T & \boldsymbol{H} \\ \boldsymbol{E}\boldsymbol{R}_{11}\boldsymbol{\Phi}^T & -\boldsymbol{I}+\boldsymbol{E}\boldsymbol{R}_{11}\boldsymbol{E}^T & \mathbf{0} \\ \boldsymbol{H}^T & \mathbf{0} & -\gamma^2\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{N}}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \prec 0 \qquad (5.63)$$

$$\begin{bmatrix} \boldsymbol{P}_{h11} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{R}_{h11} \end{bmatrix} \succeq 0, \quad \begin{bmatrix} \boldsymbol{P}_{v11} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{R}_{v11} \end{bmatrix} \succeq 0 \qquad (5.64)$$

where $\boldsymbol{\mathcal{N}}_1$ *and* $\boldsymbol{\mathcal{N}}_2$ *are full column rank matrices whose images satisfy*

$$\mathrm{Im}(\boldsymbol{\mathcal{N}}_1) = \ker(\boldsymbol{C}_2^T), \ \mathrm{Im}(\boldsymbol{\mathcal{N}}_2) = \ker([\boldsymbol{B}_2^T \boldsymbol{E}_3^T])$$

Proof. *Based on the proof of Theorem 4.7 it can be shown that the closed-loop stability along the pass holds in this case if*

$$\begin{bmatrix} -\overline{\boldsymbol{P}} & \overline{\boldsymbol{P}\boldsymbol{A}} & \overline{\boldsymbol{P}\boldsymbol{H}} & \mathbf{0} \\ \overline{\boldsymbol{A}}^T\overline{\boldsymbol{P}} & -\overline{\boldsymbol{P}} & \mathbf{0} & \overline{\boldsymbol{E}}^T \\ \overline{\boldsymbol{H}}^T\overline{\boldsymbol{P}} & \mathbf{0} & -\gamma^2\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \overline{\boldsymbol{E}} & \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \prec 0$$

where $\overline{\boldsymbol{A}}$, $\overline{\boldsymbol{H}}$, $\overline{\boldsymbol{E}}$ *are defined as before. Next, apply similar transformations to those used in the previous proof to obtain (5.62)–(5.64) and the proof is complete.* ■

To increase robustness, the term γ in the LMIs of (5.62)–(5.63) has to be minimized. This can be achieved by using a linear objective minimization procedure

$$\min_{\boldsymbol{P}_{11}\succ 0, \boldsymbol{R}_{11}\succ 0} \mu$$
$$\text{subject to } (5.62) - (5.64) \text{ with } \mu = \gamma^2$$

5.6. $\mathcal{H}_2$ control of differential LRPs

The $\mathcal{H}_2$ norm of the system is another commonly used control performance measure for control synthesis. For 2-D linear discrete systems, the $\mathcal{H}_2$ control problem was considered and solved in (Tuan *et al.*, 2002). However, the $\mathcal{H}_2$ control problem in the case of differential LRPs is still unsolved and therefore, this problem is investigated in this section.

It has been mentioned that the $\mathcal{H}_2$ norm of a process G_{diff} (5.2) is the energy (L_2 norm) of the response $g(k,t)$ to an impuls applied at $t = 0,\ k = 0$, and denoted by $\delta(k,t)$. Then (5.7) is equivalent to (by invoking Parseval's theorem in the 2-D case)

$$\|G_{\text{diff}}\|_2 = \sqrt{\|g(k,t)\|_2^2} = \sqrt{\sum_{k=0}^{\infty}\int_0^{\infty} g^T(k,t)g(k,t)dt} \qquad (5.65)$$

Note that the above equation is valid for the case of a single input stable along the pass process G_{diff}. To extend this definition to vector-valued inputs, introduce

$$u_k^h(t) = \delta(k,t)e^h$$

where $e^h \in \mathbb{R}^{l\times 1}$ is the vector whose entries are zero except for a unit entry in position h, $1 \le h \le l$. Then we have that

$$\|G_{\text{diff}}\|_2 = \sqrt{\sum_{h=1}^{l}\sum_{k=0}^{\infty}\int_0^{\infty}(g^h)^T(k,t)g^h(k,t)dt}$$

To determine $g^h(k,t)$, first introduce

$$\xi^h(k,t) = \begin{bmatrix} \dot{x}_{k+1}^h(t) \\ y_{k+1}^h(t) \end{bmatrix},\ \zeta^h(k,t) = \begin{bmatrix} x_{k+1}^h(t) \\ y_k^h(t) \end{bmatrix},\ \boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{D} \end{bmatrix},\ \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix}$$

Then, due to physical applications of LRPs where nonzero boundary conditions (2.15) appear, we have that

$$\xi^h(k,t) = (\widehat{\boldsymbol{A}}_1 + \widehat{\boldsymbol{A}}_2)\zeta^h(k,t) + \boldsymbol{\Omega}\delta(k,t)e^h = \begin{cases} \widehat{\boldsymbol{\Omega}}_h + \widehat{\boldsymbol{A}}_2\zeta^h(k,t), & \text{for } k=0, \alpha \ge t \ge 0 \\ (\widehat{\boldsymbol{A}}_1 + \widehat{\boldsymbol{A}}_2)\zeta^h(k,t), & \text{for } k>0, \alpha \ge t \ge 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.66}$$

and

$$g^h(k,t) = \boldsymbol{\Psi}\zeta^h(k,t) + \boldsymbol{D}\delta(k,t)e^h = \begin{cases} \widehat{\boldsymbol{D}}_h + \boldsymbol{D}_0 y_k^h(t), & \text{for } k=0, \alpha \ge t \ge 0 \\ \boldsymbol{\Psi}\zeta^h(k,t), & \text{for } k>0, \alpha \ge t \ge 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.67}$$

where $\widehat{\boldsymbol{D}}_h$ and $\widehat{\boldsymbol{\Omega}}_h$ denote h-th column of the matrices $\boldsymbol{D}$ and $\boldsymbol{\Omega}$ respectively. The following result gives a sufficient condition for stability along the pass and an upper bound on the $\mathcal{H}_2$ norm of the 2-D transfer function matrix.

Theorem 5.12. *A differential LRP described by (5.2) but without disturbance input is stable along the pass and has the $\mathcal{H}_2$ norm bound $\gamma > 0$, i.e. $\|G_{\text{diff}}\|_2 < \gamma$, if there exist matrices $\boldsymbol{P}_1 \succ 0$ and $\boldsymbol{P}_2 \succ 0$ such that the following LMIs hold*

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A} + \boldsymbol{C}^T\boldsymbol{C} & \boldsymbol{P}_1\boldsymbol{B}_0 + \boldsymbol{C}^T\boldsymbol{D}_0 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 + \boldsymbol{D}_0^T\boldsymbol{C} & -\boldsymbol{P}_2 + \boldsymbol{D}_0^T\boldsymbol{D}_0 \end{bmatrix} \prec 0 \tag{5.68}$$

and

$$\begin{aligned} &\text{trace}\left(\alpha\boldsymbol{D}^T\boldsymbol{D} + \alpha\boldsymbol{B}^T\boldsymbol{P}_1\boldsymbol{B} + \alpha\boldsymbol{D}^T\boldsymbol{P}_2\boldsymbol{D}\right) + \text{trace}\left(\boldsymbol{\Psi}^T\boldsymbol{P}_2\boldsymbol{\Psi}\int_0^{\alpha} f(t)f(t)^T dt\right) \\ &+ \text{trace}\left(\boldsymbol{D}_0^T\boldsymbol{D}_0\int_0^{\alpha} f(t)f(t)^T dt\right) < \gamma^2 \end{aligned} \tag{5.69}$$

Proof. *It is straightforward to see that if (5.68) holds then the following*

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A} & \boldsymbol{P}_1\boldsymbol{B}_0 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & -\boldsymbol{P}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{C}^T \\ \boldsymbol{D}_0^T \end{bmatrix}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{C} & \boldsymbol{D}_0 \end{bmatrix} \prec 0 \tag{5.70}$$

holds too. As the second term of the left hand side of (5.70) is clearly non-negative definite, it is obvious that

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A} & \boldsymbol{P}_1\boldsymbol{B}_0 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & -\boldsymbol{P}_2 \end{bmatrix} \prec 0 \tag{5.71}$$

which invoking Theorem 4.1 for process without uncertainty implies that the process is stable along the pass.

To show the $\mathcal{H}_2$ performance, recall first that (5.71) holds if $x_{k+1}(t) \neq 0$ and $y_k(t) \neq 0$ in

$$V(k,t) = \dot{x}_{k+1}^T(t)\boldsymbol{P}_1 x_{k+1}(t) + x_{k+1}^T(t)\boldsymbol{P}_1\dot{x}_{k+1}(t) + y_{k+1}^T(t)\boldsymbol{P}_2 y_{k+1}(t) - y_k^T(t)\boldsymbol{P}_2 y_k(t)$$

the following is satisfied

$$\Delta V(k,t) < 0$$

Next, introduce

$$\begin{aligned} \Delta V^h(k,t) &= (\dot{x}_{k+1}^h)^T(t)\boldsymbol{P}_1 x_{k+1}^h(t) + (x_{k+1}^h)^T(t)\boldsymbol{P}_1\dot{x}_{k+1}^h(t) \\ &\quad + (y_{k+1}^h)^T(t)\boldsymbol{P}_2 y_{k+1}^h(t) - (y_k^h)^T(t)\boldsymbol{P}_2 y_k^h(t) \end{aligned} \tag{5.72}$$

and note that

$$\Delta V(k,t) = \sum_{h=1}^{l} \Delta V^h(k,t)$$

where

$$\Delta V^h(k,t) = (\zeta^h)^T(k,t)\left(\widehat{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\widehat{\boldsymbol{A}}_1 + \widehat{\boldsymbol{A}}_2^T\boldsymbol{R}\widehat{\boldsymbol{A}}_2-\boldsymbol{R}\right)\zeta^h(k,t)$$

with $\boldsymbol{P}$, $\boldsymbol{R}$ given in (4.12) and $\widehat{\boldsymbol{A}}_1$, $\widehat{\boldsymbol{A}}_2$ defined in (4.6).

If stability along the pass holds then the following equality holds

$$\sum_{k=0}^{\infty}\int_0^{\infty} \Delta V^h(k,t) = \sum_{k=0}^{\infty}\int_0^{\alpha} \Delta V^h(k,t) = 0 \tag{5.73}$$

Furthermore, based on (5.66) and (5.72) we have

$$\begin{aligned} \sum_{k=0}^{\infty}\int_0^{\infty}\Delta V^h(k,t) &= \int_0^{\alpha}\widehat{\boldsymbol{\Omega}}_h^T(\boldsymbol{P}+\boldsymbol{R})\widehat{\boldsymbol{\Omega}}_h dt + \int_0^{\alpha}(\zeta^h)^T(0,t)\widehat{\boldsymbol{A}}_2^T(\boldsymbol{P}+\boldsymbol{R})\widehat{\boldsymbol{A}}_2\zeta^h(0,t)dt \\ &\quad + \sum_{k=0}^{\infty}\int_0^{\infty}(\zeta^h)^T(k,t)\left(\widehat{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\widehat{\boldsymbol{A}}_1+\widehat{\boldsymbol{A}}_2^T\boldsymbol{R}\widehat{\boldsymbol{A}}_2-\boldsymbol{R}\right)\zeta^h(k,t) \end{aligned}$$

Given (5.65), we also have that

$$\begin{aligned}\|G_{\text{diff}}\|_2^2 =& \sum_{h=1}^{l}\sum_{k=0}^{\infty}\int_0^{\infty} g^{hT}(k,t)g^h(k,t)dt\\ =& \sum_{h=1}^{l}\Bigg(\int_0^{\alpha}\widehat{\boldsymbol{D}}_h^T\widehat{\boldsymbol{D}}_h dt + \int_0^{\alpha}(y_0^h)^T(t)\boldsymbol{D}_0^T\boldsymbol{D}_0 y_0^h(t)dt\\ &+\sum_{k=0}^{\infty}\int_0^{\infty}\zeta^{hT}(k,t)\boldsymbol{\Psi}^T\boldsymbol{\Psi}\zeta^h(k,t)dt\Bigg)\end{aligned}$$

and also, using (5.73),

$$\begin{aligned}\|G_{\text{diff}}\|_2^2 =& \sum_{h=1}^{l}\Bigg(\sum_{k=0}^{\infty}\int_0^{\infty}\Delta V^h(k,t)dt + \alpha\widehat{\boldsymbol{D}}_h^T\widehat{\boldsymbol{D}}_h + \int_0^{\alpha}(y_0^h)^T(t)\boldsymbol{D}_0^T\boldsymbol{D}_0 y_0^h(t)dt\\ &+\sum_{k=0}^{\infty}\int_0^{\infty}\zeta^{hT}(k,t)\boldsymbol{\Psi}^T\boldsymbol{\Psi}\zeta^h(k,t)dt\Bigg)\end{aligned} \tag{5.74}$$

Routine manipulations show that (5.74) is equivalent to

$$\begin{aligned}\|G_{\text{diff}}\|_2^2 =& \sum_{h=1}^{l}\Bigg(\alpha\widehat{\boldsymbol{D}}_h^T\widehat{\boldsymbol{D}}_h + \alpha\boldsymbol{B}_h^T\boldsymbol{P}_1\boldsymbol{B}_h + \alpha\boldsymbol{D}_h^T\boldsymbol{P}_2\boldsymbol{D}_h\\ &+\int_0^{\alpha}(\zeta^h)^T(0,t)\boldsymbol{\Psi}^T\boldsymbol{P}_2\boldsymbol{\Psi}\zeta^h(0,t)dt + \int_0^{\alpha}(y_0^h)^T(t)\boldsymbol{D}_0^T\boldsymbol{D}_0 y_0^h(t)dt\\ &+\sum_{k=0}^{\infty}\int_0^{\infty}(\zeta^h)^T(k,t)\Big(\boldsymbol{\Psi}^T\boldsymbol{\Psi}+\widehat{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\widehat{\boldsymbol{A}}_1+\widehat{\boldsymbol{A}}_2^T\boldsymbol{S}\widehat{\boldsymbol{A}}_2-\boldsymbol{R}\Big)\zeta^h(k,t)dt\Bigg)\end{aligned} \tag{5.75}$$

Further transformations lead to

$$\begin{aligned}\|G_{\text{diff}}\|_2^2 =& \text{trace}\Big(\alpha\boldsymbol{D}^T\boldsymbol{D} + \alpha\boldsymbol{B}^T\boldsymbol{P}_1\boldsymbol{B} + \alpha\boldsymbol{D}^T\boldsymbol{P}_2\boldsymbol{D}\Big)\\ &+\text{trace}\left(\boldsymbol{\Psi}^T\boldsymbol{P}_2\boldsymbol{\Psi}\int_0^{\alpha} f(t)f(t)^T dt\right) + \text{trace}\left(\boldsymbol{D}_0^T\boldsymbol{D}_0\int_0^{\alpha} f(t)f(t)^T dt\right)\\ &+\sum_{k=0}^{\infty}\int_0^{\infty}\zeta^T(k,t)\Big(\boldsymbol{\Psi}^T\boldsymbol{\Psi}+\widehat{\boldsymbol{A}}_1^T\boldsymbol{P}+\boldsymbol{P}\widehat{\boldsymbol{A}}_1+\widehat{\boldsymbol{A}}_2^T\boldsymbol{S}\widehat{\boldsymbol{A}}_2-\boldsymbol{R}\Big)\zeta(k,t)dt\end{aligned}$$

It now follows immediately from this last expression that (5.68) and (5.69) imply that $\|G_{\text{diff}}\|_2 < \gamma$ holds and the proof is complete. ■

Remark 5.4. *Note that the $\mathcal{H}_2$ norm bound here can be minimized using the following linear objective minimization procedure*

$$\begin{aligned}&\min_{\boldsymbol{P}_1\succ 0,\boldsymbol{P}_2\succ 0} \quad \mu\\ &\text{subject to (5.68) and (5.69) with } \mu = \gamma^2\end{aligned} \tag{5.76}$$

5.6.1. $\mathcal{H}_2$ control with a static feedback controller

Some applications areas clearly require the design of control laws which guarantee stability along the pass and also have the maximum possible disturbance attenuation (here as measured by the $\mathcal{H}_2$ norm).

The problem considered here is as follows: for a given $\gamma > 0$, find a controller of the form (4.17) for the process (5.2) such that the closed-loop process is stable and the $\mathcal{H}_2$ norm of the 2-D transfer function matrix between the disturbance vector and the current pass profile, denoted here by $G(s, z)$ (see, (5.5)) bounded by γ, i.e. $||G(s, z)||_2 < \gamma$ - is also termed the $\mathcal{H}_2$ disturbance rejection bound.

The following result gives a solution to this problem with an algorithm for designing the control law.

Theorem 5.13. *Suppose that a differential LRP described by (5.2) is subject to a control law defined by (4.17). Then the resulting closed-loop process is stable along the pass and has the prescribed $\mathcal{H}_2$ disturbance rejection bound $\gamma > 0$ if there exist matrices $\mathbf{W}_1 \succ 0$, $\mathbf{W}_2 \succ 0$, $\mathbf{N}_1$ and $\mathbf{N}_2$ such that the following LMIs hold*

$$\begin{bmatrix} -\mathbf{W}_2 & \mathbf{C}\mathbf{W}_1+\mathbf{D}\mathbf{N}_1 & \mathbf{D}_0\mathbf{W}_2+\mathbf{D}\mathbf{N}_2 & \mathbf{0} \\ \mathbf{N}_1^T\mathbf{D}^T+\mathbf{W}_1\mathbf{C}^T & \mathbf{W}_1\mathbf{A}^T+\mathbf{A}\mathbf{W}_1+\mathbf{N}_1^T\mathbf{B}^T+\mathbf{B}\mathbf{N}_1 & \mathbf{B}_0\mathbf{W}_2+\mathbf{B}\mathbf{N}_2 & \mathbf{W}_1\mathbf{C}^T \\ \mathbf{N}_2^T\mathbf{D}^T+\mathbf{W}_2\mathbf{D}_0^T & \mathbf{W}_2\mathbf{B}_0^T+\mathbf{N}_2^T\mathbf{B}^T & -\mathbf{W}_2 & \mathbf{W}_2\mathbf{D}_0^T \\ \mathbf{0} & \mathbf{C}\mathbf{W}_1 & \mathbf{D}_0\mathbf{W}_2 & -\mathbf{I} \end{bmatrix} \prec 0 \tag{5.77}$$

and

$$\begin{gathered} \operatorname{trace}(\mathbf{X})+\operatorname{trace}(\mathbf{D}_0^T\mathbf{D}_0\boldsymbol{\Upsilon}+\alpha\mathbf{D}_1^T\mathbf{D}_1) < \gamma^2 \\ \begin{bmatrix} \mathbf{X} & \mathbf{B}_1^T & \mathbf{D}_1^T & \boldsymbol{\Pi}^{\frac{1}{2}} \\ \mathbf{B}_1 & \alpha^{-1}\mathbf{W}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_1 & \mathbf{0} & \alpha^{-1}\mathbf{W}_2 & \mathbf{0} \\ \boldsymbol{\Pi}^{\frac{1}{2}} & \mathbf{0} & \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \succ 0, \end{gathered} \tag{5.78}$$

where $\mathbf{X}$ is an additional symmetric matrix variable of proper dimension and

$$\boldsymbol{\Pi} = \int_0^\alpha \boldsymbol{\Psi}^T y_0(t) y_0^T(t) \boldsymbol{\Psi} dt, \ \ \boldsymbol{\Upsilon} = \int_0^\alpha y_0(t) y_0^T(t) dt \tag{5.79}$$

If these conditions hold, the controller matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are given by (4.20).

Proof. *Interpreting (5.75) (or Theorem 5.12 results) for the closed-loop process yields*

$$\begin{bmatrix} -\mathbf{P}_2 & \mathbf{P}_2\mathbf{C}+\mathbf{P}_2\mathbf{D}\mathbf{K}_1 & \mathbf{P}_2\mathbf{D}_0+\mathbf{P}_2\mathbf{D}\mathbf{K}_2 \\ \mathbf{K}_1^T\mathbf{D}^T\mathbf{P}_2+\mathbf{C}^T\mathbf{P}_2 & \boldsymbol{\Lambda}_1 & \mathbf{P}_1\mathbf{B}_0+\mathbf{P}_1\mathbf{B}\mathbf{K}_1+\mathbf{C}^T\mathbf{D}_0 \\ \mathbf{K}_2^T\mathbf{D}^T\mathbf{P}_2+\mathbf{D}_0^T\mathbf{P}_2 & \mathbf{B}_0^T\mathbf{P}_1+\mathbf{K}_2^T\mathbf{B}^T\mathbf{P}_1+\mathbf{D}_0^T\mathbf{C} & -\mathbf{P}_2+\mathbf{D}_0^T\mathbf{D}_0 \end{bmatrix} \prec 0$$

where $\boldsymbol{\Omega}^T = \begin{bmatrix} \mathbf{B}_1^T & \mathbf{D}_1^T \end{bmatrix}$ and $\boldsymbol{\Lambda}_1 = \mathbf{A}^T\mathbf{P}_1+\mathbf{P}_1\mathbf{A}+\mathbf{K}_1^T\mathbf{B}^T\mathbf{P}_1+\mathbf{P}_1\mathbf{B}\mathbf{K}_1+\mathbf{C}^T\mathbf{C}$. Now set $\mathbf{W}_1 = \mathbf{P}_1^{-1}$, $\mathbf{W}_2 = \mathbf{P}_2^{-1}$, and pre- and post- multiply both sides of this *last inequality by* $\operatorname{diag}(\boldsymbol{W}_2, \boldsymbol{W}_1, \boldsymbol{W}_2)$ *to obtain*

$$\begin{bmatrix} -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1 + \boldsymbol{D}\boldsymbol{K}_1\boldsymbol{W}_1 & \boldsymbol{D}_0\boldsymbol{W}_2 + \boldsymbol{D}\boldsymbol{K}_2\boldsymbol{W}_2 \\ \boldsymbol{W}_1\boldsymbol{K}_1^T\boldsymbol{D}^T + \boldsymbol{W}_1\boldsymbol{C}^T & \boldsymbol{\Lambda}_2 & \boldsymbol{\Lambda}_3^T \\ \boldsymbol{W}_2\boldsymbol{K}_2^T\boldsymbol{D}^T + \boldsymbol{W}_2\boldsymbol{D}_0^T & \boldsymbol{\Lambda}_3 & -\boldsymbol{W}_2 + \boldsymbol{W}_2\boldsymbol{D}_0^T\boldsymbol{D}_0\boldsymbol{W}_2 \end{bmatrix} \prec 0$$

where

$$\begin{aligned} \boldsymbol{\Lambda}_2 &= \boldsymbol{W}_1\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{W}_1 + \boldsymbol{W}_1\boldsymbol{K}_1^T\boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{K}_1\boldsymbol{W}_1 + \boldsymbol{W}_1\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{W}_1 \\ \boldsymbol{\Lambda}_3 &= \boldsymbol{W}_2\boldsymbol{B}_0^T + \boldsymbol{W}_2\boldsymbol{K}_2^T\boldsymbol{B}^T + \boldsymbol{W}_2\boldsymbol{D}_0^T\boldsymbol{C}\boldsymbol{W}_1 \end{aligned}$$

An obvious application of the Schur complement formula to the left hand side of this last expression and setting $\boldsymbol{N}_1 = \boldsymbol{K}_1\boldsymbol{W}_1$ *and* $\boldsymbol{N}_2 = \boldsymbol{K}_2\boldsymbol{W}_2$ *now yields*

$$\begin{bmatrix} -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1 + \boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2 + \boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{0} \\ \boldsymbol{N}_1^T\boldsymbol{D}^T + \boldsymbol{W}_1\boldsymbol{C}^T & \boldsymbol{W}_1\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{W}_1 + \boldsymbol{N}_1^T\boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2 + \boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{W}_1\boldsymbol{C}^T \\ \boldsymbol{N}_2^T\boldsymbol{D}^T + \boldsymbol{W}_2\boldsymbol{D}_0^T & \boldsymbol{W}_2\boldsymbol{B}_0^T + \boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 & \boldsymbol{W}_2\boldsymbol{D}_0^T \\ \boldsymbol{0} & \boldsymbol{C}\boldsymbol{W}_1 & \boldsymbol{D}_0\boldsymbol{W}_2 & -\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.80}$$

In what follows, by observing the fact that

$$\operatorname{trace}(\boldsymbol{P}_2\boldsymbol{\Pi}) = \operatorname{trace}\left(\boldsymbol{\Pi}^{\frac{1}{2}}\boldsymbol{P}_2\boldsymbol{\Pi}^{\frac{1}{2}}\right)$$

where $\boldsymbol{\Pi}$ *is defined in (5.79) then the inequality (5.69) in this case becomes*

$$\begin{aligned} &\operatorname{trace}(\alpha\boldsymbol{D}_1^T\boldsymbol{D}_1 + \boldsymbol{D}_0^T\boldsymbol{D}_0\boldsymbol{\Upsilon}) \\ &+\operatorname{trace}\left(\begin{bmatrix} \boldsymbol{B}_1^T & \boldsymbol{D}_1^T & \boldsymbol{\Pi}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \alpha\boldsymbol{W}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \alpha\boldsymbol{W}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{D}_1 \\ \boldsymbol{\Pi}^{\frac{1}{2}} \end{bmatrix}\right) < \gamma^2 \end{aligned} \tag{5.81}$$

which is equivalent to (5.78). To see this, introduce a new matrix variable $\boldsymbol{X}$ *and make use of the following transformation to yield*

$$\boldsymbol{\Omega}^T\boldsymbol{\Theta}\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{X} - \alpha\boldsymbol{B}_1^T\boldsymbol{W}_1^{-1}\boldsymbol{B}_1 - \alpha\boldsymbol{D}_1^T\boldsymbol{W}_2^{-1}\boldsymbol{D}_1 - \boldsymbol{\Pi}^{\frac{1}{2}}\boldsymbol{W}_2\boldsymbol{\Pi}^{\frac{1}{2}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 \end{bmatrix}$$

where

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{X} & \alpha^{\frac{1}{2}}\boldsymbol{B}_1^T & \alpha^{\frac{1}{2}}\boldsymbol{D}_1^T & \boldsymbol{\Pi}^{\frac{1}{2}} \\ \alpha^{\frac{1}{2}}\boldsymbol{B}_1 & \boldsymbol{W}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \alpha^{\frac{1}{2}}\boldsymbol{D}_1 & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} \\ \boldsymbol{\Pi}^{\frac{1}{2}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 \end{bmatrix}, \quad \boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -\alpha^{\frac{1}{2}}\boldsymbol{W}_1^{-1}\boldsymbol{B}_1 & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ -\alpha^{\frac{1}{2}}\boldsymbol{W}_2^{-1}\boldsymbol{D}_1 & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{W}_2^{-1}\boldsymbol{\Pi}^{\frac{1}{2}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

Next, observe that the block (1,1) in the resulting matrix implies that

$$\boldsymbol{X} \succ \alpha\boldsymbol{B}_1^T\boldsymbol{W}_1^{-1}\boldsymbol{B}_1 + \alpha\boldsymbol{D}_1^T\boldsymbol{W}_2^{-1}\boldsymbol{D}_1 + \boldsymbol{\Pi}^{\frac{1}{2}}\boldsymbol{W}_2^{-1}\boldsymbol{\Pi}^{\frac{1}{2}}$$

Finally, apply the Schur complement formula and the proof is complete. ■

Remark 5.5. *The $\mathcal{H}_2$ disturbance rejection bound γ in the LMI of (5.78) can be minimized by using the linear objective minimization procedure*

$$\begin{aligned} &\min_{\boldsymbol{W}_1 \succ 0, \boldsymbol{W}_2 \succ 0, \boldsymbol{N}_1, \boldsymbol{N}_2} \quad \mu \\ &\text{subject to (5.77) and (5.78) with } \mu = \gamma^2 \end{aligned} \tag{5.82}$$

Let us now show the applicability of this result with the following numerical example.

Example 5.2. *Consider the differential LRP represented by (a model of metal rolling process, for details see (Gałkowski et al., 2003d))*

$$\boldsymbol{A} = \begin{bmatrix} -0.0050 & -5.8077 \\ 1 & -0.0050 \end{bmatrix}, \ \boldsymbol{B}_0 = \begin{bmatrix} 0 \\ 0.0494 \end{bmatrix}, \ \boldsymbol{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \ \boldsymbol{D}_0 = 0.7692 \tag{5.83}$$

where the boundary condition are

$$x_{k+1}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \ y_0(t) = 1, \ \ 0 \le t \le \alpha$$

Further, we assume the matrices $\boldsymbol{B}$, $\boldsymbol{D}$, $\boldsymbol{B}_1$ and $\boldsymbol{D}_1$ to be

$$\boldsymbol{B} = \begin{bmatrix} 1.2 \\ 0.8 \end{bmatrix}, \ \boldsymbol{D} = 1.2, \ \boldsymbol{B}_1 = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}, \ \boldsymbol{D}_1 = 0.8$$

The purpose is to find a controller of the form (4.17) such that (5.77) and (5.78) are satisfied. By solving the convex optimization problem (5.82) we obtain

$$\begin{aligned} \boldsymbol{W}_1 &= \begin{bmatrix} 2.6102 & 0.2420 \\ 0.2420 & 0.0657 \end{bmatrix}, \ \boldsymbol{W}_2 = 0.7052, \\ \boldsymbol{N}_1 &= \begin{bmatrix} -2.7623 & -0.5935 \end{bmatrix}, \ \boldsymbol{N}_2 = -0.4522 \end{aligned}$$

Then, the following controller matrices are computed

$$\boldsymbol{K}_1 = \begin{bmatrix} -0.3353 & -7.7998 \end{bmatrix}, \ \boldsymbol{K}_2 = -0.6412$$

That is, the controller guarantees that the closed-loop process is stable along the pass and ensures that the $\mathcal{H}_2$ norm bound is never greater than 7.6607. *The results have demonstrated that the proposed approach is numerically simple and effective.*

5.7. Guaranteed cost control of LRPs

In previous sections of this chapter, the problem of designing $\mathcal{H}_\infty$ or $\mathcal{H}_2$ controllers has been addressed. Here, another step is taken further to examine the next design approach called guaranteed cost control (Moheimani and Petersen, 1996; Petersen and McFarlane, 1994; Petersen *et al.*, 1998). This is an area for which no results currently exist and here we develop a solution to the problem of obtaining a control law which simultaneously robustly stabilises an uncertain process and guarantees that the associated cost function has a value below the prescribed upper bound. Keeping up with this objective throughout the section, we will start by treating differential LRPs and then deal with discrete LRPs.

5.7.1. Differential LRP case

Consider an uncertain differential LRP described by the following state-space model over $0 \le t \le \alpha$, $k \ge 0$

$$\begin{aligned}\dot{x}_{k+1}(t) =&(\boldsymbol{A}+\Delta\boldsymbol{A})x_{k+1}(t)+(\boldsymbol{B}_0+\Delta\boldsymbol{B}_0)y_k(t)+(\boldsymbol{B}+\Delta\boldsymbol{B})u_{k+1}(t)\\ y_{k+1}(t) =&(\boldsymbol{C}+\Delta\boldsymbol{C})x_{k+1}(t)+(\boldsymbol{D}_0+\Delta\boldsymbol{D}_0)y_k(t)+(\boldsymbol{D}+\Delta\boldsymbol{D})u_{k+1}(t)\end{aligned} \tag{5.84}$$

The matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{B}_0$, $\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{D}_0$ define the nominal model and $\Delta\boldsymbol{A}$, $\Delta\boldsymbol{B}$, $\Delta\boldsymbol{B}_0$, $\Delta\boldsymbol{C}$, $\Delta\boldsymbol{D}$, $\Delta\boldsymbol{D}_0$ represent admissible uncertainties which are assumed to be of the form

$$\begin{bmatrix}\Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0 & \Delta\boldsymbol{B}\\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 & \Delta\boldsymbol{D}\end{bmatrix}=\begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\end{bmatrix}\boldsymbol{\mathcal{F}}\begin{bmatrix}\boldsymbol{E}_1 & \boldsymbol{E}_2 & \boldsymbol{E}_3\end{bmatrix} \tag{5.85}$$

In this last equation, $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ are known constant matrices of compatible dimensions, and $\boldsymbol{\mathcal{F}}$ is an unknown matrix with constant entries which satisfies (4.5).

We start by developing the LMI condition which guarantees that the unforced (the control input terms are deleted) process is stable along the pass and also the associated cost function is bounded for all admissible uncertainties. These results are then extended to design a guaranteed cost controller.

It is assumed that the following cost function is associated with the uncertain process (5.84)

$$J=\sum_{k=0}^{\infty}\int_0^{\infty}\left(u_{k+1}^T(t)\boldsymbol{\Psi}u_{k+1}(t)\right)dt+\sum_{k=0}^{\infty}\int_0^{\infty}\left(\begin{bmatrix}x_{k+1}(t)\\ y_k(t)\end{bmatrix}^T\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_2\end{bmatrix}\begin{bmatrix}x_{k+1}(t)\\ y_k(t)\end{bmatrix}\right)dt \tag{5.86}$$

where $\boldsymbol{\Psi}\succ 0$, $\boldsymbol{Q}_1\succ 0$ and $\boldsymbol{Q}_2\succ 0$ are given matrices, is bounded for all admissible uncertainties.

Remark 5.6. *LRPs are defined over the finite pass length α and, in practice, only a finite number of passes, say k^*, will actually be completed. Hence the cost function (5.86) should be replaced by*

$$J=\sum_{k=0}^{k^*}\int_0^{\alpha}\left(u_{k+1}^T(t)\boldsymbol{\Psi}u_{k+1}(t)\right)dt+\sum_{k=0}^{k^*}\int_0^{\alpha}\left(\begin{bmatrix}x_{k+1}(t)\\ y_k(t)\end{bmatrix}^T\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_2\end{bmatrix}\begin{bmatrix}x_{k+1}(t)\\ y_k(t)\end{bmatrix}\right)dt \tag{5.87}$$

However, it is routine to argue that the signals involved can be extended from $[0,\alpha]$ to the infinite interval in such a way that projection of the infinite interval solution onto the finite interval is possible. The same is true for the pass-to-pass direction and hence we can work with (5.86).

5.7.1.1. Guaranteed cost bound

Here we are interested in finding an upper bound for the corresponding cost function of the unforced process ($u_{k+1}(t) = 0$) with associated cost function

$$J_0 = \sum_{k=0}^{\infty} \int_0^{\infty} \left(\begin{bmatrix} x_{k+1}(t) \\ y_k(t) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{Q}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_2 \end{bmatrix} \begin{bmatrix} x_{k+1}(t) \\ y_k(t) \end{bmatrix} \right) dt \tag{5.88}$$

The following theorem gives a sufficient condition for stability along the pass with guaranteed cost.

Theorem 5.14. *An unforced differential LRP described by (5.84) is robustly stable if there exist matrices $\boldsymbol{P}_1 \succ 0$, $\boldsymbol{P}_2 \succ 0$ and a scalar $\epsilon > 0$ such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A} + \boldsymbol{Q}_1 + \epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \boldsymbol{P}_1\boldsymbol{B}_0 & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & -\boldsymbol{P}_2 + \boldsymbol{Q}_2 + \epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \mathbf{0} & -\epsilon\boldsymbol{I} & \mathbf{0} \\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.89}$$

Moreover, in this case the cost function (5.88) satisfies the following upper bound

$$J_0 \le \sum_{k=0}^{k^*} x_{k+1}^T(0)\boldsymbol{P}_1 x_{k+1}(0) + \int_0^{\alpha} y_0^T(t)\boldsymbol{P}_2 y_0(t)dt \tag{5.90}$$

Proof. *Based on Theorem 4.1 proof, it is shown that stability along the pass holds if $\Delta V(k,t) < 0$ (defined in (4.11)) for $\xi(k,t) \neq 0$. Next, it is straightforward to see that the inequality*

$$\Delta V(k,t) + \xi^T(k,t) \begin{bmatrix} \boldsymbol{Q}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_2 \end{bmatrix} \xi(k,t) \prec 0 \tag{5.91}$$

implies that unforced process is stable along the pass. Noting that

$$\Upsilon = \sum_{k=0}^{\infty} \int_0^{\infty} \left(\xi^T(k,t) \begin{bmatrix} \boldsymbol{Q}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_2 \end{bmatrix} \xi(k,t) \right) dt$$

and, since the process is stable along the pass, we now have that

$$\begin{aligned} \Upsilon \le & -\sum_{k=0}^{\infty} \int_0^{\infty} (\dot{V}_1(k,t) + \Delta V_2(k,t))dt \\ = & -\sum_{k=0}^{\infty} x_{k+1}^T(t)\boldsymbol{P}_1 x_{k+1}(t) \Big|_0^{\infty} - \int_0^{\infty} \left(\sum_{k=0}^{\infty} (y_{k+1}^T(t)\boldsymbol{P}_2 y_{k+1}(t) - y_k^T(t)\boldsymbol{P}_2 y_k(t)) \right) dt \\ = & \sum_{k=0}^{\infty} x_{k+1}^T(0)\boldsymbol{P}_1 x_{k+1}(0) - \int_0^{\infty} \Big(y_{\infty}^T(t)\boldsymbol{P}_2 y_{\infty}(t) - y_0^T(t)\boldsymbol{P}_2 y_0(t) \Big) dt \\ = & \sum_{k=0}^{\infty} x_{k+1}^T(0)\boldsymbol{P}_1 x_{k+1}(0) + \int_0^{\infty} y_0^T(t)\boldsymbol{P}_2 y_0(t)dt \end{aligned} \tag{5.92}$$

Using (4.11) and (5.91), a sufficient condition for stability along the pass which ensures that (5.90) holds is given by

$$\big((\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)^T\boldsymbol{P}+\boldsymbol{P}(\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)+(\boldsymbol{A}_2+\Delta\boldsymbol{A}_2)\boldsymbol{S}(\boldsymbol{A}_2+\Delta\boldsymbol{A}_2)-\boldsymbol{R}+\boldsymbol{Q}\big)\prec 0$$

where $\boldsymbol{Q}=\mathrm{diag}(\boldsymbol{Q}_1,\boldsymbol{Q}_2)$, $\boldsymbol{S}=\mathrm{diag}(\boldsymbol{P}_3,\boldsymbol{P}_2)$, *and* $\boldsymbol{P}_3\succ 0$ *are any given matrices of the required dimensions. Next, an obvious application of the Schur complement formula yields*

$$\begin{bmatrix} -\boldsymbol{P}_3 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C}+\boldsymbol{P}_2\Delta\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0+\boldsymbol{P}_2\Delta\boldsymbol{D}_0\\ \boldsymbol{0} & \boldsymbol{C}^T\boldsymbol{P}_2+\Delta\boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{\Lambda}_1 & \boldsymbol{P}_1\boldsymbol{B}_0+\boldsymbol{P}_1\Delta\boldsymbol{B}_0\\ \boldsymbol{0} & \boldsymbol{D}_0^T\boldsymbol{P}_2+\Delta\boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1+\Delta\boldsymbol{B}_0^T\boldsymbol{P}_1 & \boldsymbol{Q}_2-\boldsymbol{P}_2 \end{bmatrix}\prec 0 \qquad (5.93)$$

where

$$\boldsymbol{\Lambda}_1=\boldsymbol{A}^T\boldsymbol{P}_1+\Delta\boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A}+\boldsymbol{P}_1\Delta\boldsymbol{A}+\boldsymbol{Q}_1$$

On removing the block $-\boldsymbol{P}_3$ *which is always negative definite, (5.93) gives the equivalent condition*

$$\begin{aligned}&\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0\\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A}+\boldsymbol{Q}_1 & \boldsymbol{P}_1\boldsymbol{B}_0\\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & \boldsymbol{Q}_2-\boldsymbol{P}_2 \end{bmatrix}\\ &+\begin{bmatrix} \boldsymbol{0} & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2\\ \boldsymbol{0} & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix} \boldsymbol{\mathcal{F}} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\mathcal{F}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{F}}\end{bmatrix}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{E}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2\end{bmatrix}\\ &+\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{E}_1^T & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2^T\end{bmatrix}\begin{bmatrix} \boldsymbol{\mathcal{F}}^T & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\mathcal{F}}^T & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{F}}^T\end{bmatrix}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{0}\\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{0}\end{bmatrix}\prec 0\end{aligned} \qquad (5.94)$$

and by an obvious application of the result of Lemma 8, we obtain

$$\begin{aligned}&\begin{bmatrix} -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0\\ \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{P}_1\boldsymbol{A}+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1+\boldsymbol{Q}_1 & \boldsymbol{P}_1\boldsymbol{B}_0\\ \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{B}_0^T\boldsymbol{P}_1 & \boldsymbol{Q}_2-\boldsymbol{P}_2+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 \end{bmatrix}\\ &+\epsilon^{-1}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2\\ \boldsymbol{0} & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{0}\\ \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{0}\end{bmatrix}\prec 0\end{aligned} \qquad (5.95)$$

Finally, an obvious application of the Schur complement formula gives (5.89) and the proof is complete. ■

Remark 5.7. *Note that it is possible to minimize the upper bound on the cost function (5.90) using the following optimization procedure*

$$\begin{aligned}&\min_{\boldsymbol{P}_1\succ 0,\boldsymbol{P}_2\succ 0}\left[\sum_{k=0}^{k^*}x_{k+1}^T(0)\boldsymbol{P}_1x_{k+1}(0)+\int_0^{\alpha}y_0^T(t)\boldsymbol{P}_2y_0(t)dt\right]\\ &\text{subject to}:\ \ (5.89)\end{aligned} \qquad (5.96)$$

5.7.1.2. Guaranteed cost control with a static feedback controller

Now we proceed with the design of a static controller of the form of (4.17) that stabilises the process (5.84) and guarantees the cost is bounded.

Applying the control law (4.17) to (5.84) gives the closed-loop process state-space model of the form (4.18) and the associated cost function is

$$J=\sum_{k=0}^{\infty}\int_0^{\infty}\left(\begin{bmatrix} x_{k+1}(t)\\ y_k(t)\end{bmatrix}^T\begin{bmatrix}\boldsymbol{Q}_1+\boldsymbol{K}_1^T\boldsymbol{\Psi}\boldsymbol{K}_1 & \boldsymbol{K}_1^T\boldsymbol{\Psi}\boldsymbol{K}_2\\ \boldsymbol{K}_2^T\boldsymbol{\Psi}\boldsymbol{K}_1 & \boldsymbol{Q}_2+\boldsymbol{K}_2^T\boldsymbol{\Psi}\boldsymbol{K}_2\end{bmatrix}\begin{bmatrix} x_{k+1}(t)\\ y_k(t)\end{bmatrix}\right)dt \tag{5.97}$$

where $\boldsymbol{\Psi}\succ 0$, $\boldsymbol{Q}_1\succ 0$ and $\boldsymbol{Q}_2\succ 0$ are given matrices.

The existence of stabilising $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ can be characterized in LMI terms as follows.

Theorem 5.15. *A differential LRP described by (5.84) is robustly stable under the control law (4.17) if there exist matrices $\boldsymbol{W}_1\succ 0$, $\boldsymbol{W}_2\succ 0$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ and a scalar $\epsilon>0$ such that the following LMI holds*

$$\left[\begin{array}{cccccccc}
-\boldsymbol{W}_2+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_1^T & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{B}\boldsymbol{N}_1+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_1^T & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \boldsymbol{0} & \boldsymbol{N}_1^T & \boldsymbol{W}_1 & \boldsymbol{0}\\
\boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2\boldsymbol{E}_3^T & \boldsymbol{N}_2^T & \boldsymbol{0} & \boldsymbol{W}_2\\
\boldsymbol{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{N}_1 & \boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{\Psi}^{-1} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{W}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Q}_1^{-1} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Q}_2^{-1}
\end{array}\right]\prec 0 \tag{5.98}$$

where $\boldsymbol{\Psi}\succ 0$, $\boldsymbol{Q}_1\succ 0$ and $\boldsymbol{Q}_2\succ 0$ are the given matrices for the cost function (5.86). Also, if this condition holds, then stabilising control law matrices $\boldsymbol{K}_1$, $\boldsymbol{K}_2$ are given by (4.20). The cost function (5.97) of the closed-loop process (4.18) satisfies the following upper bound

$$J\le\sum_{k=0}^{k^*}x_{k+1}^T(0)\boldsymbol{W}_1^{-1}x_{k+1}(0)+\int_0^{\alpha}y_0^T(t)\boldsymbol{W}_2^{-1}y_0(t)dt \tag{5.99}$$

Proof. *Since (5.89) is satisfied then applying the some matrix manipulations i.e. setting $\boldsymbol{W}_1=\boldsymbol{P}_1^{-1}$, $\boldsymbol{W}_2=\boldsymbol{P}_2^{-1}$, $\boldsymbol{U}_1=\boldsymbol{W}_1\boldsymbol{Q}_1\boldsymbol{W}_1$, $\boldsymbol{U}_2=\boldsymbol{W}_2\boldsymbol{Q}_2\boldsymbol{W}_2$ followed by pre- and post- multiplying of both sides of resulting inequality by* $\mathrm{diag}(\boldsymbol{W}_2,\boldsymbol{W}_1,\boldsymbol{W}_2)$ *we conclude that the closed-loop process (4.18) is robustly stabilised by the control* *law (4.17) if the following matrix inequality is satisfied*

$$\begin{bmatrix} -\boldsymbol{W}_2 & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 \\ \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & \boldsymbol{\Upsilon} & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2+\boldsymbol{N}_1\boldsymbol{\Psi}\boldsymbol{N}_2^T \\ \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2^T\boldsymbol{D}^T & \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T+\boldsymbol{N}_2^T\boldsymbol{\Psi}\boldsymbol{N}_1 & -\boldsymbol{W}_2+\boldsymbol{U}_2+\boldsymbol{N}_2^T\boldsymbol{\Psi}\boldsymbol{N}_2 \end{bmatrix}$$
$$+\begin{bmatrix} \mathbf{0} & \boldsymbol{H}_2 & \boldsymbol{H}_2 \\ \mathbf{0} & \boldsymbol{H}_1 & \boldsymbol{H}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}} \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 \end{bmatrix}$$
$$+\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{N}_1^T\boldsymbol{E}_3^T+\boldsymbol{W}_2\boldsymbol{E}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{N}_2^T\boldsymbol{E}_3^T+\boldsymbol{W}_2\boldsymbol{E}_2^T \end{bmatrix}\begin{bmatrix} \boldsymbol{\mathcal{F}}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}}^T \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_2^T & \boldsymbol{H}_1^T & \mathbf{0} \\ \boldsymbol{H}_2^T & \boldsymbol{H}_1^T & \mathbf{0} \end{bmatrix} \prec 0 \tag{5.100}$$

where $\boldsymbol{N}_1 = \boldsymbol{K}_1\boldsymbol{W}_1$, $\boldsymbol{N}_2 = \boldsymbol{K}_2\boldsymbol{W}_2$ *and* $\boldsymbol{\Upsilon} = \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{N}_1^T\boldsymbol{B}^T+\boldsymbol{B}\boldsymbol{N}_1+\boldsymbol{U}_1+\boldsymbol{N}_1^T\boldsymbol{\Psi}\boldsymbol{N}_1$. *Applying the result of Lemma 8 and making an obvious application of the Schur complement formula gives (5.98) and the proof is complete.* ■

Note that it is possible to minimize the upper bound on the cost function (5.90) using the following optimization procedure

$$\min_{\boldsymbol{W}_1\succ 0, \boldsymbol{W}_2\succ 0}\left[\sum_{k=0}^{k^*} x_{k+1}^T(0)\boldsymbol{W}_1^{-1}x_{k+1}(0)+\int_0^{\alpha} y_0^T(t)\boldsymbol{W}_2^{-1}y_0(t)dt\right] \tag{5.101}$$

subject to (5.98)

The convex optimization algorithm cannot be applied in this case because of the nonlinear terms $\boldsymbol{W}_1^{-1}$ and $\boldsymbol{W}_2^{-1}$. However, a controller which ensures the minimization of the guaranteed cost (5.99) can be achieved as follows. First note that, from the fact that trace$(\boldsymbol{X}\boldsymbol{Y})$ = trace$(\boldsymbol{Y}\boldsymbol{X})$, we have

$$\begin{aligned}\sum_{k=0}^{k^*} x_{k+1}^T(0)\boldsymbol{W}_1^{-1}x_{k+1}(0) &= \sum_{k=0}^{k^*} \text{trace}\left(x_{k+1}^T(0)\boldsymbol{W}_1^{-1}x_{k+1}(0)\right) \\ &= \sum_{k=0}^{k^*} \text{trace}\left(\boldsymbol{W}_1^{-1}x_{k+1}(0)x_{k+1}^T(0)\right)\end{aligned}$$

and

$$\int_0^{\alpha} y_0^T(t)\boldsymbol{W}_2^{-1}y_0(t)dt = \int_0^{\alpha}\text{trace}\left(y_0^T(t)\boldsymbol{W}_2^{-1}y_0(t)\right) = \int_0^{\alpha}\text{trace}\left(\boldsymbol{W}_2^{-1}y_0(t)y_0^T(t)\right)$$

Next, recall that if a matrix $\boldsymbol{M}$ is symmetric and positive semi-definite i.e. $\boldsymbol{M} \succeq 0$ then the eigenvalue decomposition of such a matrix gives

$$\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Theta}\boldsymbol{V}^T \tag{5.102}$$

where $\boldsymbol{V}$ is some unitary matrix and $\boldsymbol{\Theta}$ is a diagonal with nonnegative diagonal entries. Therefore, the matrix square root of $\boldsymbol{M}$ can be defined as $\boldsymbol{M}^{\frac{1}{2}} = \boldsymbol{V}\boldsymbol{\Theta}^{\frac{1}{2}}\boldsymbol{V}^T$

and computed (Golub and Loan, 1996). Based on this, the matrices $\boldsymbol{\Upsilon}^{\frac{1}{2}}$ and $\boldsymbol{\Sigma}^{\frac{1}{2}}$

$$\begin{aligned}\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}^{\frac{1}{2}}\boldsymbol{\Upsilon}^{\frac{1}{2}} &= \sum_{k=0}^{k^*} x_{k+1}(0)x_{k+1}^T(0)\\ \boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{\Sigma}^{\frac{1}{2}} &= \int_0^{\alpha} y_0(t)y_0^T(t)dt\end{aligned}$$

can be obtained. The dimensions of $\boldsymbol{\Sigma}^{\frac{1}{2}}$ and $\boldsymbol{\Upsilon}^{\frac{1}{2}}$ are $n \times n$ and $m \times m$ respectively. Furthermore, introduce the symmetric matrices $\boldsymbol{\Xi}$, $\boldsymbol{\Omega}$ which satisfy

$$\begin{aligned}\text{trace}(\boldsymbol{\Upsilon}^{\frac{1}{2}}\boldsymbol{W}_1^{-1}\boldsymbol{\Upsilon}^{\frac{1}{2}}) &< \text{trace}(\boldsymbol{\Xi})\\ \text{trace}(\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{W}_2^{-1}\boldsymbol{\Sigma}^{\frac{1}{2}}) &< \text{trace}(\boldsymbol{\Omega})\end{aligned}$$

hence we can write

$$\boldsymbol{\Upsilon}^{\frac{1}{2}}\boldsymbol{W}_1^{-1}\boldsymbol{\Upsilon}^{\frac{1}{2}} \prec \boldsymbol{\Xi}, \quad \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{W}_2^{-1}\boldsymbol{\Sigma}^{\frac{1}{2}} \prec \boldsymbol{\Omega} \tag{5.103}$$

Carrying out an obvious application of the Schur complement of (5.103) yields

$$\begin{bmatrix} -\boldsymbol{\Xi} & \boldsymbol{\Upsilon}^{\frac{1}{2}} \\ \boldsymbol{\Upsilon}^{\frac{1}{2}} & -\boldsymbol{W}_1 \end{bmatrix} \prec 0 \quad \text{and} \quad \begin{bmatrix} -\boldsymbol{\Omega} & \boldsymbol{\Sigma}^{\frac{1}{2}} \\ \boldsymbol{\Sigma}^{\frac{1}{2}} & -\boldsymbol{W}_2 \end{bmatrix} \prec 0 \tag{5.104}$$

respectively. Finally, the following minimization problem can be formulated as

$$\begin{aligned}\min_{\boldsymbol{W}_1\succ 0, \boldsymbol{W}_2 \succ 0, \boldsymbol{N}_1, \boldsymbol{N}_2} \quad & (\text{trace}(\boldsymbol{\Xi}) + \text{trace}(\boldsymbol{\Omega}))\\ \text{subject to}: \quad & (5.98) \text{ and } (5.104)\end{aligned} \tag{5.105}$$

and the solution (4.20) now guarantees that the cost function is minimized over the finite pass length in the case when only a finite number of trials is actually completed. Since the minimization problem of (5.105) is the convex optimization problem, then it is simple to implement using a computer and computationally effective.

To show the validity of the above controller design procedure, let us provide the following numerical example.

Example 5.3. *Consider the differential LRP represented by (5.83) with the following boundary conditions*

$$x_{k+1}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \ y_0(t) = 1, \ \ 0 \leq t \leq \alpha$$

Suppose also that the matrices $\boldsymbol{B}$, $\boldsymbol{D}$, $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$ and $\boldsymbol{E}_3$ are

$$\begin{aligned}\boldsymbol{B} = \begin{bmatrix} 1.2 \\ 0.8 \end{bmatrix}, \ \boldsymbol{D} = 1.2, \ \boldsymbol{H}_1 = \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}, \ \boldsymbol{H}_2 = 0.1,\\ \boldsymbol{E}_1 = 1.0 \cdot 10^{-3} \begin{bmatrix} 0.3 & 0.2 \end{bmatrix}, \ \boldsymbol{E}_2 = 0.2, \ \boldsymbol{E}_3 = 0.5\end{aligned} \tag{5.106}$$

Further, it is assumed that

$$\boldsymbol{Q}_1 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \ \boldsymbol{Q}_2 = 20, \ \boldsymbol{\Psi} = 10$$

then the application of procedure (5.105) for 20 *passes (i.e.* $k^* = 20$*) and* $\alpha = 10$ *gives the solution matrices*

$$\boldsymbol{W}_1 = \begin{bmatrix} 0.0256 & -0.0017 \\ -0.0017 & 0.0022 \end{bmatrix}, \ \boldsymbol{W}_2 = 0.02249, \ \boldsymbol{\Sigma} = 444.5218,$$
$$\boldsymbol{N}_1 = \begin{bmatrix} -0.0199 & -0.0025 \end{bmatrix}, \ \boldsymbol{N}_2 = -0.0050, \ \ \epsilon = 0.0069$$

Hence we get the following controller matrices

$$\boldsymbol{K}_1 = \left[\ -0.8965 \quad -1.8275 \ \right], \ \boldsymbol{K}_2 = -0.2201$$

The obtained controller guarantees the stability along the pass of the closed-loop process and ensures cost bound (5.99). It is clear that upper cost bound is 444.5218 *i.e. it is equal to* trace $(\boldsymbol{\Sigma})$. *On the other hand, it follows from (5.99) that*

$$\int_0^{10} y_0^T(t)\boldsymbol{W}_2^{-1}y_0(t)dt = 10 \cdot \boldsymbol{W}_2^{-1} = 444.642$$

This means that the proposed numerical procedure for controller matrices design provides an effective method to obtain the minimum cost.

5.7.2. Discrete LRP case

The problem of designing a controller for a discrete LRP to make the closed-loop process robustly stable and to minimize a quadratic cost has not yet been considered in any paper. However, the solution to this problem for a clear discrete 2-D system has been presented in (Guan *et al.*, 2001) therefore it can be the basis for developing the result in terms of a discrete LRP. Furthermore, due to the fact that in terms of LRPs, the pass profile vector is simultaneously the output vector, this leads to some simplification in relation to a clear 2-D approach. This allows us to design the dynamic pass profile controller with the use of Lemma 9 what is novel to the known results (Guan *et al.*, 2001).

Let us consider the following state-space model of discrete LRP

$$\begin{aligned} x_{k+1}(p+1) &= (\boldsymbol{A}+\Delta\boldsymbol{A})x_{k+1}(p)+(\boldsymbol{B}_0+\Delta\boldsymbol{B}_0)y_k(p)+(\boldsymbol{B}+\Delta\boldsymbol{B})u_{k+1}(p) \\ y_{k+1}(p) &= (\boldsymbol{C}+\Delta\boldsymbol{C})x_{k+1}(p)+(\boldsymbol{D}_0+\Delta\boldsymbol{D}_0)y_k(p)+(\boldsymbol{D}+\Delta\boldsymbol{D})u_{k+1}(p) \end{aligned} \tag{5.107}$$

The matrices $\Delta\boldsymbol{A}$, $\Delta\boldsymbol{B}$, $\Delta\boldsymbol{B}_0$, $\Delta\boldsymbol{C}$, $\Delta\boldsymbol{D}$, $\Delta\boldsymbol{D}_0$ represent admissible uncertainties to be of the form

$$\begin{bmatrix} \Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0 & \Delta\boldsymbol{B} \\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 & \Delta\boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} \boldsymbol{\mathcal{F}} \left[\ \boldsymbol{E}_1 \ \boldsymbol{E}_2 \ \boldsymbol{E}_3 \ \right] \tag{5.108}$$

where $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ are some known constant matrices with compatible dimensions and $\boldsymbol{\mathcal{F}}$ is an unknown constant matrix which satisfies (4.5).

Associated with the uncertain process (5.107) is the cost function

$$J=\sum_{k=0}^{\infty}\sum_{p=0}^{\infty}\left(u_{k+1}^T(p)\boldsymbol{\Psi}u_{k+1}(p)\right)+\sum_{k=0}^{\infty}\sum_{p=0}^{\infty}\left(\begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix}^T\begin{bmatrix} \boldsymbol{Q}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_2 \end{bmatrix}\begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix}\right) \tag{5.109}$$

where $\boldsymbol{\Psi} \succ 0$, $\boldsymbol{Q}_1 \succ 0$ and $\boldsymbol{Q}_2 \succ 0$ are design matrices to be specified. This cost function is bounded for all admissible uncertainties. In physical terms this cost function can be interpreted as the sum of quadratic costs on the input, state and pass profile vectors on each pass.

The approach taken in this section is as follows: we first derive a sufficient condition which guarantees that the unforced (the control input terms are deleted) process is stable along the pass with an associated cost function which is bounded for all admissible uncertainties and then this result is extended to design a guaranteed cost controller in both the static and dynamic version.

Remark 5.8. *It is significant to note that a discrete LRP is defined over the finite pass length α and, in practice, only a finite number of passes, say k^*, will actually be completed. Hence, the cost function bound is computed over finite intervals $p \in [0,\alpha]$ and $k \in [0,k^*]$. However, in theoretic operations the infinite interval in both directions (i.e. along a given pass and from pass to pass directions) are considered - see Remark 5.6.*

5.7.2.1. Guaranteed cost bound

Since the process is assumed to be unforced ($u_{k+1}(p) = 0$) then the associated cost function (5.117) becomes

$$J_0=\sum_{k=0}^{\infty}\sum_{p=0}^{\infty}\left(\begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix}^T\begin{bmatrix} \boldsymbol{Q}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_2 \end{bmatrix}\begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix}\right) \tag{5.110}$$

The following theorem gives a sufficient condition for stability along the pass with a guaranteed cost.

Theorem 5.16. *An unforced discrete LRP described by (5.107) is stable along the pass for all admissible uncertainties if there exist matrices $\boldsymbol{P}_1 \succ 0$, $\boldsymbol{P}_2 \succ 0$ and a scalar $\epsilon > 0$ such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{P}_1 & \mathbf{0} & \boldsymbol{P}_1\boldsymbol{A} & \boldsymbol{P}_1\boldsymbol{B}_0 & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \mathbf{0} & -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \boldsymbol{A}^T\boldsymbol{P}_1 & \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{Q}_1-\boldsymbol{P}_1+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{B}_0^T\boldsymbol{P}_1 & \boldsymbol{D}_0^T\boldsymbol{P}_2 & \mathbf{0} & \boldsymbol{Q}_2-\boldsymbol{P}_2+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.111}$$

Also if this condition holds, the cost function (5.110) satisfies the upper bound

$$J_0 \leq \sum_{k=0}^{k^*} x_{k+1}(0)\boldsymbol{P}_1 x_{k+1}(0) + \sum_{p=0}^{\alpha} y_0^T(p)\boldsymbol{P}_2 y_0(p) \tag{5.112}$$

Proof. *First, note that the inequality*

$$\Delta V(k,p) + \zeta^T(k,p)\boldsymbol{Q}\zeta(k,p) < 0 \tag{5.113}$$

implies that unforced process (5.107) is stable along the pass where $\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{Q}_1, \boldsymbol{Q}_2)$. *Combining (4.35) and (5.113) yields the following sufficient condition for stability along the pass*

$$(\boldsymbol{A}_1 + \Delta\boldsymbol{A}_1)^T \boldsymbol{P}(\boldsymbol{A}_1 + \Delta\boldsymbol{A}_1) + (\boldsymbol{A}_2 + \Delta\boldsymbol{A}_2)\boldsymbol{P}(\boldsymbol{A}_2 + \Delta\boldsymbol{A}_2) - \boldsymbol{P} + \boldsymbol{Q} \prec 0 \tag{5.114}$$

Now suppose that stability along the pass holds and introduce

$$\Upsilon = \sum_{k=0}^{\infty}\sum_{p=0}^{\infty} \zeta^T(k,p)\boldsymbol{Q}\zeta(k,p)$$

then

$$\begin{aligned}
\Upsilon \leq & -\sum_{k=0}^{\infty}\sum_{p=0}^{\infty}(\Delta V_1(k,p) + \Delta V_2(k,p)) \\
= & -\sum_{k=0}^{\infty}\left(\sum_{p=0}^{\infty} x_{k+1}(p+1)^T \boldsymbol{P}_1 x_{k+1}(p+1) - x_{k+1}^T(p)\boldsymbol{P}_1 x_{k+1}(p)\right) \\
& -\sum_{p=0}^{\infty}\left(\sum_{k=0}^{\infty} y_{k+1}^T(p)\boldsymbol{P}_2 y_{k+1}(p) - y_k^T(p)\boldsymbol{P}_2 y_k(p)\right) \\
= & \sum_{k=0}^{\infty} x_{k+1}^T(0)\boldsymbol{P}_1 x_{k+1}(0) + \sum_{p=0}^{\infty} y_0^T(p)\boldsymbol{P}_2 y_0(p)
\end{aligned}$$

which ensures that (5.112) holds. Next, application of the Schur complement formula to inequality (5.114) followed by using the result of Lemma 8 yields

$$\begin{bmatrix} -\boldsymbol{P}_1 & \boldsymbol{0} & \boldsymbol{P}_1\boldsymbol{A} & \boldsymbol{P}_1\boldsymbol{B}_0 \\ \boldsymbol{0} & -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C} & \boldsymbol{P}_2\boldsymbol{D}_0 \\ \boldsymbol{A}^T\boldsymbol{P}_1 & \boldsymbol{C}^T\boldsymbol{P}_2 & \boldsymbol{Q}_1 - \boldsymbol{P}_1 + \epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1 & \boldsymbol{0} \\ \boldsymbol{B}_0^T\boldsymbol{P}_1 & \boldsymbol{D}_0^T\boldsymbol{P}_2 & \boldsymbol{0} & \boldsymbol{Q}_2 - \boldsymbol{P}_2 + \epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 \end{bmatrix}$$
$$+\ \epsilon^{-1}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \prec 0$$

Finally, application of the Schur complement formula to this last expression gives (5.111) and the proof is complete. ■

Note that it is possible to minimize the upper bound on the cost function (5.112) using the following optimization procedure

$$\min_{\boldsymbol{P}_1\succ 0,\boldsymbol{P}_2\succ 0}\left[\sum_{k=0}^{k^*} x_{k+1}^T(0)\boldsymbol{P}_1 x_{k+1}(0)+\sum_{p=0}^{\alpha} y_0^T(p)\boldsymbol{P}_2 y_0(p)\right]$$
$$\text{subject to}\quad (5.89)$$

5.7.2.2. Guaranteed cost control with a static feedback controller

Here, it is assumed that all states are available for feedbacks then the control law of the form (4.40) can be applied to a process described by (5.107). Hence the associated cost function for the resulting closed-loop process is given by

$$J=\sum_{k=0}^{\infty}\sum_{p=0}^{\infty}\left(\begin{bmatrix} x_{k+1}(p)\\ y_k(p)\end{bmatrix}^T\begin{bmatrix}\boldsymbol{Q}_1+\boldsymbol{K}_1^T\boldsymbol{\Psi}\boldsymbol{K}_1 & \boldsymbol{K}_1^T\boldsymbol{\Psi}\boldsymbol{K}_2\\ \boldsymbol{K}_2^T\boldsymbol{\Psi}\boldsymbol{K}_1 & \boldsymbol{Q}_2+\boldsymbol{K}_2^T\boldsymbol{\Psi}\boldsymbol{K}_2\end{bmatrix}\begin{bmatrix} x_{k+1}(p)\\ y_k(p)\end{bmatrix}\right) \tag{5.115}$$

which is of the form of that in Theorem 5.16 and we have the following result.

Theorem 5.17. *Suppose that a control law of the form (4.40) is applied to a discrete LRP described by (5.107). Then the resulting closed-loop process is stable along the pass for all admissible uncertainties if there exist matrices $\boldsymbol{W}_1\succ 0$, $\boldsymbol{W}_2\succ 0$, $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ and a scalar $\epsilon>0$ such that the following LMI holds*

$$\begin{bmatrix}
-\boldsymbol{W}_1+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_1^T & 2\epsilon\boldsymbol{H}_2\boldsymbol{H}_1^T & \boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
2\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & -\boldsymbol{W}_2+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T & \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & -\boldsymbol{W}_1 & \boldsymbol{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \boldsymbol{0} & \boldsymbol{N}_1^T & \boldsymbol{W}_1 & \boldsymbol{0}\\
\boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2\boldsymbol{D}^T & \boldsymbol{0} & -\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T & \boldsymbol{N}_2^T & \boldsymbol{0} & \boldsymbol{W}_2\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{N}_1 & \boldsymbol{N}_2 & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{\Psi}^{-1} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Q}_1^{-1} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Q}_2^{-1}
\end{bmatrix}\prec 0 \tag{5.116}$$

Also, if this condition holds, then stabilising control law matrices $\boldsymbol{K}_1$, $\boldsymbol{K}_2$ are given by (4.20) and the cost function (5.115) of the closed-loop process satisfies the following upper bound

$$J\leq\sum_{k=0}^{k^*} x_{k+1}^T(0)\boldsymbol{W}_1^{-1}x_{k+1}(0)+\sum_{p=0}^{\alpha} y_0^T(p)\boldsymbol{W}_2^{-1}y_0(p) \tag{5.117}$$

Proof. *Based on interpreting (5.89) for the state-space model considered here, we conclude that the closed-loop process is robustly stabilised by the control law (4.40) if the following matrix inequality is satisfied*

$$\begin{bmatrix} -\boldsymbol{P}_1 & \mathbf{0} & \boldsymbol{P}_1\boldsymbol{A}+\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{K}_1 & \boldsymbol{P}_1\boldsymbol{B}_0+\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{K}_2 \\ \mathbf{0} & -\boldsymbol{P}_2 & \boldsymbol{P}_2\boldsymbol{C}+\boldsymbol{P}_2\boldsymbol{D}\boldsymbol{K}_1 & \boldsymbol{P}_2\boldsymbol{D}_0+\boldsymbol{P}_2\boldsymbol{D}\boldsymbol{K}_2 \\ \boldsymbol{A}^T\boldsymbol{P}_1+\boldsymbol{K}_1^T\boldsymbol{B}^T\boldsymbol{P}_1 & \boldsymbol{C}^T\boldsymbol{P}_2+\boldsymbol{K}_1^T\boldsymbol{D}^T\boldsymbol{P}_2 & \boldsymbol{Q}_1-\boldsymbol{P}_1+\boldsymbol{K}_1^T\boldsymbol{\Psi}\boldsymbol{K}_1 & \boldsymbol{K}_1^T\boldsymbol{\Psi}\boldsymbol{K}_2 \\ \boldsymbol{B}_0^T\boldsymbol{P}_1+\boldsymbol{K}_2^T\boldsymbol{B}^T\boldsymbol{P}_1 & \boldsymbol{D}_0^T\boldsymbol{P}_2+\boldsymbol{K}_2\boldsymbol{D}^T\boldsymbol{P}_1 & \boldsymbol{K}_2^T\boldsymbol{\Psi}\boldsymbol{K}_1 & \boldsymbol{Q}_2-\boldsymbol{P}_2+\boldsymbol{K}_2^T\boldsymbol{\Psi}\boldsymbol{K}_2 \end{bmatrix}$$
$$+\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_1^T+\boldsymbol{K}_1^T\boldsymbol{E}_3^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2^T+\boldsymbol{K}_2^T\boldsymbol{E}_3^T \end{bmatrix}\begin{bmatrix} \boldsymbol{\mathcal{F}}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}}^T \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{H}_1^T\boldsymbol{P}_1 & \boldsymbol{H}_2^T\boldsymbol{P}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$
$$+\begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{P}_1\boldsymbol{H}_1 & \boldsymbol{P}_1\boldsymbol{H}_1 \\ \mathbf{0} & \mathbf{0} & \boldsymbol{P}_2\boldsymbol{H}_2 & \boldsymbol{P}_2\boldsymbol{H}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}} \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_1+\boldsymbol{E}_3\boldsymbol{K}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2+\boldsymbol{E}_3\boldsymbol{K}_2 \end{bmatrix} \prec 0$$

Now set $\boldsymbol{W}_1 = \boldsymbol{P}_1^{-1}$, $\boldsymbol{W}_2 = \boldsymbol{P}_2^{-1}$, $\boldsymbol{U}_1 = \boldsymbol{W}_1\boldsymbol{Q}_1\boldsymbol{W}_1$ *and* $\boldsymbol{U}_2 = \boldsymbol{W}_2\boldsymbol{Q}_2\boldsymbol{W}_2$ *and then pre- and post- multiply both sides of this last inequality by* $\mathrm{diag}(\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_1, \boldsymbol{W}_2)$. *Next, apply the result of Lemma 8 to obtain*

$$\begin{bmatrix} -\boldsymbol{W}_1+2\epsilon\boldsymbol{H}_1\boldsymbol{H}_1^T & 2\epsilon\boldsymbol{H}_2\boldsymbol{H}_1^T & \boldsymbol{A}\boldsymbol{W}_1+\boldsymbol{B}\boldsymbol{N}_1 & \boldsymbol{B}_0\boldsymbol{W}_2+\boldsymbol{B}\boldsymbol{N}_2 \\ 2\epsilon\boldsymbol{H}_1\boldsymbol{H}_2^T & -\boldsymbol{W}_2+2\epsilon\boldsymbol{H}_2\boldsymbol{H}_2^T & \boldsymbol{C}\boldsymbol{W}_1+\boldsymbol{D}\boldsymbol{N}_1 & \boldsymbol{D}_0\boldsymbol{W}_2+\boldsymbol{D}\boldsymbol{N}_2 \\ \boldsymbol{W}_1\boldsymbol{A}^T+\boldsymbol{N}_1^T\boldsymbol{B}^T & \boldsymbol{W}_1\boldsymbol{C}^T+\boldsymbol{N}_1^T\boldsymbol{D}^T & \boldsymbol{U}_1-\boldsymbol{W}_1+\boldsymbol{N}_1^T\boldsymbol{\Psi}\boldsymbol{N}_1 & \boldsymbol{N}_1^T\boldsymbol{\Psi}\boldsymbol{N}_2 \\ \boldsymbol{W}_2\boldsymbol{B}_0^T+\boldsymbol{N}_2^T\boldsymbol{B}^T & \boldsymbol{W}_2\boldsymbol{D}_0^T+\boldsymbol{N}_2\boldsymbol{D}^T & \boldsymbol{N}_2^T\boldsymbol{\Psi}\boldsymbol{N}_1 & \boldsymbol{U}_2-\boldsymbol{W}_2+\boldsymbol{N}_2^T\boldsymbol{\Psi}\boldsymbol{N}_2 \end{bmatrix}$$
$$+\epsilon^{-1}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{W}_1\boldsymbol{E}_1^T+\boldsymbol{N}_1^T\boldsymbol{E}_3^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{W}_2\boldsymbol{E}_2^T+\boldsymbol{N}_2^T\boldsymbol{E}_3^T \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_1\boldsymbol{W}_1+\boldsymbol{E}_3\boldsymbol{N}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}_2+\boldsymbol{E}_3\boldsymbol{N}_2 \end{bmatrix} \prec 0$$

where $\boldsymbol{N}_1 = \boldsymbol{K}_1\boldsymbol{W}_1$ *and* $\boldsymbol{N}_2 = \boldsymbol{K}_2\boldsymbol{W}_2$. *Finally, making an obvious application of the Schur complement formula gives (5.116) and* $\boldsymbol{P}_1 = \boldsymbol{W}_1^{-1}$ *and* $\boldsymbol{P}_2 = \boldsymbol{W}_2^{-1}$, *(5.112) is converted into (5.117). Finally, the bound on the cost function (5.117) can be established in an identical manner to that on* J_0 *in the previous result. Hence the details are omitted here.* ■

The presence of the nonlinear terms $\boldsymbol{W}_1^{-1}$ and $\boldsymbol{W}_2^{-1}$ in (5.117) means that it is not possible to apply a linear objective minimization procedure to minimize the cost function (5.117). However, a controller which ensures the minimization of the guaranteed cost can be achieved as follows. First, use the eigenvalue decomposition (5.102) to compute the matrices $\boldsymbol{\Sigma}_1^{\frac{1}{2}}$ and $\boldsymbol{\Sigma}_2^{\frac{1}{2}}$ (i.e. the matrix square roots) which satisfy

$$\boldsymbol{\Sigma}_1^{\frac{1}{2}}\boldsymbol{\Sigma}_1^{\frac{1}{2}} = \sum_{k=0}^{k^*} x_{k+1}^T(0)x_{k+1}(0), \quad \boldsymbol{\Sigma}_2^{\frac{1}{2}}\boldsymbol{\Sigma}_2^{\frac{1}{2}} = \sum_{p=0}^{\alpha} y_0^T(p)y_0(p)$$

and hence we can write

$$\sum_{k=0}^{k^*} x_{k+1}^T(0)\boldsymbol{W}_1^{-1}x_{k+1}(0) = \text{trace}(\boldsymbol{\Sigma}_1^T\boldsymbol{W}_1^{-1}\boldsymbol{\Sigma}_1)$$

$$\sum_{p=0}^{\alpha} y_0^T(p)\boldsymbol{W}_2^{-1}y_0(p) = \text{trace}(\boldsymbol{\Sigma}_2^T\boldsymbol{W}_2^{-1}\boldsymbol{\Sigma}_2)$$

Next, introduce the symmetric matrices $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ such that

$$\boldsymbol{\Sigma}_1^{\frac{1}{2}}\boldsymbol{W}_1^{-1}\boldsymbol{\Sigma}_1^{\frac{1}{2}} \prec \boldsymbol{\Omega}_1, \quad \boldsymbol{\Sigma}_2^{\frac{1}{2}}\boldsymbol{W}_2^{-1}\boldsymbol{\Sigma}_2^{\frac{1}{2}} \prec \boldsymbol{\Omega}_2 \tag{5.118}$$

Application of the Schur complement formula gives

$$\begin{bmatrix} -\boldsymbol{\Omega}_1 & \boldsymbol{\Sigma}_1^{\frac{1}{2}} \\ \boldsymbol{\Sigma}_1^{\frac{1}{2}} & -\boldsymbol{W}_1 \end{bmatrix} \prec 0 \quad \text{and} \quad \begin{bmatrix} -\boldsymbol{\Omega}_2 & \boldsymbol{\Sigma}_2^{\frac{1}{2}} \\ \boldsymbol{\Sigma}_2^{\frac{1}{2}} & -\boldsymbol{W}_2 \end{bmatrix} \prec 0 \tag{5.119}$$

Finally, the following minimization problem can be formulated as

$$\begin{aligned} &\min_{\boldsymbol{W}_1\succ 0,\boldsymbol{W}_2\succ 0,\boldsymbol{N}_1,\boldsymbol{N}_2} \quad \text{trace}(\boldsymbol{\Omega}_1+\boldsymbol{\Omega}_2) \\ &\text{subject to} \quad (5.116) \text{ and } (5.119) \end{aligned}$$

which gives a controller that guarantees the cost function is minimized.

5.7.2.3. Guaranteed cost control with a full dynamic pass profile controller

Under the assumption that the process state is completely accessible to feedback, we developed a static feedback controller that stabilises process (5.107) and guarantees an upper bound for the cost function defined by (5.117). When the process state is not available, we can use a dynamic pass profile controller to stabilise discrete LRPs and guarantee that the cost is bounded.

To simplify notation, the following matrices are introduced

$$\Delta\boldsymbol{\Phi} = \begin{bmatrix} \Delta\boldsymbol{A} & \Delta\boldsymbol{B}_0 \\ \Delta\boldsymbol{C} & \Delta\boldsymbol{D}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} \boldsymbol{\mathcal{F}} \begin{bmatrix} \boldsymbol{E}_1 & \boldsymbol{E}_2 \end{bmatrix},$$

$$\Delta\boldsymbol{B}_2 = \begin{bmatrix} \Delta\boldsymbol{B} \\ \Delta\boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} \boldsymbol{\mathcal{F}} \begin{bmatrix} \boldsymbol{E}_3 \end{bmatrix}$$

where $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ are known real matrices satisfying (5.108) and the matrix $\boldsymbol{\mathcal{F}}$ satisfies (4.5).

Substituting (5.50) into (5.107) (and assuming $\boldsymbol{D}_c = \boldsymbol{0}$) yields the resulting closed-loop process

$$\begin{aligned} \begin{bmatrix} \overline{x}_{k+1}(p+1) \\ \overline{y}_{k+1}(p) \end{bmatrix} &= (\widetilde{\boldsymbol{A}} + \Delta\widetilde{\boldsymbol{A}}) \begin{bmatrix} \overline{x}_{k+1}(p) \\ \overline{y}_k(p) \end{bmatrix} \\ y_{k+1}(p) &= \widetilde{\boldsymbol{C}} \begin{bmatrix} \overline{x}_{k+1}(p) \\ \overline{y}_k(p) \end{bmatrix} \end{aligned} \tag{5.120}$$

where

$$\begin{aligned}\widetilde{\boldsymbol{A}}+\Delta\widetilde{\boldsymbol{A}} =&\boldsymbol{\Pi}\begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{B}_c\boldsymbol{C}_2 & \boldsymbol{A}_c\end{bmatrix}\boldsymbol{\Pi}^T+\boldsymbol{\Pi}\begin{bmatrix}\Delta\boldsymbol{\Phi} & \Delta\boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Pi}^T\\ =&\boldsymbol{\Pi}\begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{B}_2\boldsymbol{C}_c\\ \boldsymbol{B}_c\boldsymbol{C}_2 & \boldsymbol{A}_c\end{bmatrix}\boldsymbol{\Pi}^T+\boldsymbol{\Pi}\begin{bmatrix}\boldsymbol{H}\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{\mathcal{F}}\begin{bmatrix}\boldsymbol{E} & \boldsymbol{E}_3\boldsymbol{C}_c\end{bmatrix}\boldsymbol{\Pi}^T\\ =&\widetilde{\boldsymbol{A}}+\overline{\boldsymbol{H}}\boldsymbol{\mathcal{F}}\overline{\boldsymbol{E}},\\ \widetilde{\boldsymbol{C}}=&\begin{bmatrix}\boldsymbol{C}_2 & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Pi}^T\end{aligned}$$

and the matrices $\boldsymbol{H}$ and $\boldsymbol{E}$ are given by (5.61). The associated cost function is

$$J=\sum_{k=0}^{\infty}\sum_{p=0}^{\infty}\left(\begin{bmatrix}\overline{x}_{k+1}(p)\\ \overline{y}_k(p)\end{bmatrix}^T\boldsymbol{\Pi}\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Y}\end{bmatrix}\boldsymbol{\Pi}^T\begin{bmatrix}\overline{x}_{k+1}(p)\\ \overline{y}_k(p)\end{bmatrix}\right) \tag{5.121}$$

where $\boldsymbol{Q}=\operatorname{diag}(\boldsymbol{Q}_1,\boldsymbol{Q}_2)$, $\boldsymbol{Y}=\boldsymbol{C}_c^T\boldsymbol{\Psi}\boldsymbol{C}_c$ and $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$, $\boldsymbol{\Psi}$ are given matrices of (5.109).

Now we have the following result which gives the existence condition for guaranteed cost controller of the form (5.50) (with $\boldsymbol{D}_c=\boldsymbol{0}$).

Theorem 5.18. *Suppose that a control law of the form (5.50) is applied to a discrete LRP of the form considered here with the associated uncertainty structure. Then the resulting closed-loop process is stable along the pass if there exist matrices $\boldsymbol{P}_{11}\succ 0$, $(\boldsymbol{P}_{11}=\operatorname{diag}(\boldsymbol{P}_{h11},\boldsymbol{P}_{v11}))$, $\boldsymbol{R}_{11}\succ 0$, $(\boldsymbol{R}_{11}=\operatorname{diag}(\boldsymbol{R}_{h11},\boldsymbol{R}_{v11}))$ such that the LMIs defined by (5.122)–(5.123) hold*

$$\begin{bmatrix}\boldsymbol{\mathcal{N}}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{\Xi}^T\boldsymbol{P}_{11}\boldsymbol{\Xi}-\boldsymbol{P}_{11} & \boldsymbol{\Xi}^T\boldsymbol{P}_{11}\boldsymbol{H} & \boldsymbol{E}^T & \boldsymbol{Q}^{\frac{1}{2}}\\ \boldsymbol{H}^T\boldsymbol{P}_{11}\boldsymbol{\Xi} & \boldsymbol{H}^T\boldsymbol{P}_{11}\boldsymbol{H}-\epsilon\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{E} & \boldsymbol{0} & -\epsilon^{-1}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{Q}^{\frac{1}{2}} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\mathcal{N}}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\prec 0 \tag{5.122}$$

$$\begin{bmatrix}\boldsymbol{\mathcal{N}}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{\Xi}\boldsymbol{R}_{11}\boldsymbol{\Xi}^T-\boldsymbol{R}_{11} & \boldsymbol{\Xi}\boldsymbol{R}_{11}\boldsymbol{E}^T & \boldsymbol{0} & \boldsymbol{H} & \boldsymbol{\Xi}\boldsymbol{R}_{11}\boldsymbol{Q}^{\frac{1}{2}}\\ \boldsymbol{E}\boldsymbol{R}_{11}\boldsymbol{\Xi}^T & -\epsilon^{-1}\boldsymbol{I}+\boldsymbol{E}\boldsymbol{R}_{11}\boldsymbol{E}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}\boldsymbol{R}_{11}\boldsymbol{Q}^{\frac{1}{2}}\\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{H}^T & \boldsymbol{0} & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{R}_{11}\boldsymbol{\Xi}^T & \boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{R}_{11}\boldsymbol{E}^T & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\mathcal{N}}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\prec 0 \tag{5.123}$$

$$\begin{bmatrix}\boldsymbol{P}_{h11} & \boldsymbol{I}\\ \boldsymbol{I} & \boldsymbol{R}_{h11}\end{bmatrix}\succ 0,\quad \begin{bmatrix}\boldsymbol{P}_{v11} & \boldsymbol{I}\\ \boldsymbol{I} & \boldsymbol{R}_{v11}\end{bmatrix}\succ 0 \tag{5.124}$$

where $\boldsymbol{\mathcal{N}}_1$ and $\boldsymbol{\mathcal{N}}_2$ are full column rank matrices whose images satisfy

$$\operatorname{Im}(\boldsymbol{\mathcal{N}}_1)=\ker(\boldsymbol{C}_2),\ \operatorname{Im}(\boldsymbol{\mathcal{N}}_2)=\ker\left(\begin{bmatrix}\boldsymbol{B}_2^T & \boldsymbol{E}_3^T & \boldsymbol{\Psi}^{\frac{1}{2}}\end{bmatrix}\right)$$

and ϵ is a given positive scalar. If these conditions hold, the cost function (5.121) of the closed-loop process (5.120) satisfies the following upper bound

$$J\leq\sum_{k=0}^{k^*}\sum_{p=0}^{\alpha}\left(\begin{bmatrix}\overline{x}_{k+1}(p)\\ \overline{y}_k(p)\end{bmatrix}^T\boldsymbol{\Pi}\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Y}\end{bmatrix}\boldsymbol{\Pi}^T\begin{bmatrix}\overline{x}_{k+1}(p)\\ \overline{y}_k(p)\end{bmatrix}\right) \tag{5.125}$$

Proof. *Following the steps in the proof of Theorem 4.7 it follows that stability along the pass condition for the uncertain process (5.120) can be written in the form*

$$\begin{bmatrix} -\boldsymbol{P} & \widetilde{\boldsymbol{A}} & \overline{\boldsymbol{H}} & \boldsymbol{0} & \boldsymbol{0} \\ \widetilde{\boldsymbol{A}}^T & -\boldsymbol{P} & \boldsymbol{0} & \overline{\boldsymbol{E}}^T & \overline{\boldsymbol{S}}^T \\ \overline{\boldsymbol{H}}^T & \boldsymbol{0} & -\epsilon \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \overline{\boldsymbol{E}} & \boldsymbol{0} & -\epsilon^{-1}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \overline{\boldsymbol{S}} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \prec 0 \tag{5.126}$$

where $\widetilde{\boldsymbol{A}}$, $\overline{\boldsymbol{H}}$, $\overline{\boldsymbol{E}}$ *are as before and*

$$\begin{aligned}\overline{\boldsymbol{S}} = &\begin{bmatrix} \boldsymbol{Q}^{\frac{1}{2}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi}^{\frac{1}{2}}\boldsymbol{C}_c \end{bmatrix} \boldsymbol{\Pi}^T = \left(\begin{bmatrix} \boldsymbol{Q}^{\frac{1}{2}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Psi}^{\frac{1}{2}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{C}_c \\ \boldsymbol{B}_c & \boldsymbol{A}_c \end{bmatrix} \begin{bmatrix} \boldsymbol{C}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \right) \boldsymbol{\Pi}^T \\ =&\widehat{\boldsymbol{Q}}\boldsymbol{\Pi}^T + \widehat{\boldsymbol{\Psi}}\boldsymbol{\Theta}\boldsymbol{\mathcal{C}}_2\boldsymbol{\Pi}^T\end{aligned}$$

and $\boldsymbol{\mathcal{C}}_2$ *and* $\boldsymbol{\Theta}$ *(with* $\boldsymbol{D}_c = \boldsymbol{0}$*) are also as before. Next, pre multiply (5.126) by* $\mathrm{diag}(\boldsymbol{\Pi}^T, \boldsymbol{\Pi}^T, \boldsymbol{I}, \boldsymbol{I}, \boldsymbol{I})$, *post-multiply it by the transpose of this last matrix, and set* $\boldsymbol{R} = \boldsymbol{\Pi}^T\boldsymbol{P}\boldsymbol{\Pi}$ *(see, (5.57)) to rewrite (5.126) as*

$$\boldsymbol{\Psi} + \boldsymbol{M}^T\boldsymbol{\Theta}\boldsymbol{N} + \boldsymbol{N}\boldsymbol{\Theta}^T\boldsymbol{M} \prec 0$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} -\boldsymbol{R} & \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{H}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{A}}^T & -\boldsymbol{R} & \boldsymbol{0} & \boldsymbol{\mathcal{E}}^T & \widehat{\boldsymbol{Q}}^T \\ \boldsymbol{\mathcal{H}}^T & \boldsymbol{0} & -\epsilon\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{E}} & \boldsymbol{0} & -\epsilon^{-1}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \widehat{\boldsymbol{Q}} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \ \boldsymbol{M}^T = \begin{bmatrix} \boldsymbol{\mathcal{B}}_2 \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{\mathcal{E}}_4 \\ \widehat{\boldsymbol{\Psi}} \end{bmatrix}, \ \boldsymbol{N} = \begin{bmatrix} \boldsymbol{0} \ \boldsymbol{\mathcal{C}}_2 \ \boldsymbol{0} \ \boldsymbol{0} \ \boldsymbol{0} \end{bmatrix}$$

and

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} \boldsymbol{H} \\ \boldsymbol{0} \end{bmatrix} \ \boldsymbol{\mathcal{E}} = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \end{bmatrix} \ \boldsymbol{\mathcal{E}}_4 = \begin{bmatrix} \boldsymbol{E}_3 & \boldsymbol{0} \end{bmatrix}$$

Since

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{\mathcal{B}}_2^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{E}}_4^T & \widehat{\boldsymbol{\Psi}}^T \end{bmatrix} = \left[\begin{array}{cc|cc|c|c|cc} \boldsymbol{B}_2^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_3^T & \boldsymbol{0} & \boldsymbol{\Psi}^{\frac{1}{2}} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{array}\right]$$

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\mathcal{C}}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} = \left[\begin{array}{cc|cc|c|c|cc} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{array}\right]$$

then the kernels of $\boldsymbol{M}$ *and* $\boldsymbol{N}$ *are the images of*

$$\boldsymbol{\mathcal{W}}_{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{\mathcal{N}}_{11} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{N}}_{12} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{\mathcal{N}}_{13} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{\mathcal{W}}_N = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{N}}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

where $\boldsymbol{\mathcal{N}}_{11} = \ker(\boldsymbol{B}_2^T)$*,* $\boldsymbol{\mathcal{N}}_{12} = \ker(\boldsymbol{E}_3^T)$*,* $\boldsymbol{\mathcal{N}}_{13} = \ker(\boldsymbol{\Psi}^{\frac{1}{2}})$ *and* $\boldsymbol{\mathcal{N}}_2 = \ker(\boldsymbol{C}_2)$*. Now invoke Lemma 9 to obtain the following conditions which are equivalent to (5.126)*

$$\boldsymbol{\mathcal{W}}_M^T \boldsymbol{\Psi} \boldsymbol{\mathcal{W}}_M \prec 0 \quad \textit{and} \quad \boldsymbol{\mathcal{W}}_N^T \boldsymbol{\Psi} \boldsymbol{\mathcal{W}}_N \prec 0$$

Since some rows of $\boldsymbol{\mathcal{W}}_M$ *and* $\boldsymbol{\mathcal{W}}_N$ *are zero then*

$$\boldsymbol{\mathcal{W}}_M = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{N}}_{11} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{N}}_{12} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{N}}_{13} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ 0 & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ 0 & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ 0 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ 0 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{\mathcal{W}}_N = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{N}}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

Routine matrix manipulations yield (5.122)–(5.123). Finally, the cost function bound is established in an identical manner to that of the previous result and hence the details are omitted here. ∎

The guaranteed cost controller here can be computed as in the previous case, see for example Section 5.2.3.

The interesting point to note is that the presented result provides the alternative computational method for designing the guaranteed cost controller by the method included in (Guan *et al.*, 2001). To illustrate the effectiveness and implementation simplicity of the proposed LMI condition for controller computation, the following example is provided.

Example 5.4. *Consider the discrete LRP as described by (4.29) and suppose that the process data are*

$$\boldsymbol{A} = \begin{bmatrix} 0.4841 & 0.0599 \\ 0.6488 & 0.5585 \end{bmatrix}, \ \boldsymbol{B}_0 = \begin{bmatrix} 0.1574 & 0.0000 \\ 0.1312 & 0.0262 \end{bmatrix}, \ \boldsymbol{C} = \begin{bmatrix} 0.0990 & 0.0455 \\ 0.0077 & 0.0656 \end{bmatrix},$$

$$\boldsymbol{D}_0 = \begin{bmatrix} 0.8508 & 0.5133 \\ 0.1863 & 0.4568 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 1.3672 & 2.5656 \\ 2.5893 & 1.4168 \end{bmatrix}, \ \boldsymbol{D} = \begin{bmatrix} 2.6984 & 0.7550 \\ 0.9412 & 1.2990 \end{bmatrix}$$

and take the matrices defining the uncertainty model as

$$\boldsymbol{H} = \begin{bmatrix} 0.0326 & 0.0884 \\ 0.0380 & 0.0457 \\ 0.0886 & 0.0799 \\ 0.0761 & 0.0134 \end{bmatrix}, \ \boldsymbol{E} = \begin{bmatrix} 0.0065 & 0.0374 & 0.0969 & 0.0253 \\ 0.0375 & 0.0484 & 0.0342 & 0.0585 \end{bmatrix},$$

$$\boldsymbol{E}_3 = \begin{bmatrix} 0.0524 & 0.0486 \\ 0.0163 & 0.0496 \end{bmatrix}$$

and the matrices $\boldsymbol{Q}_1$*,* $\boldsymbol{Q}_2$ *and* $\boldsymbol{\Psi}$ *in the cost function (5.115) as*

$$\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \begin{bmatrix} 80 & 0 \\ 0 & 80 \end{bmatrix}, \ \boldsymbol{\Psi} = 40$$

Application of the controller design procedure of Theorem 5.18 for 10 *passes and* $\alpha = 20$ *and with the boundary conditions of the form*

$$x_{k+1}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \;\; y_0(p) = 1, \;\; 0 \le p \le \alpha$$

gives the solution matrices

$$\begin{aligned}
\boldsymbol{P}_{h11} =& 10^4 \begin{bmatrix} 8.5599 & -3.7049 \\ -3.7049 & 3.0750 \end{bmatrix}, \; \boldsymbol{P}_{v11} = 10^5 \begin{bmatrix} 0.5156 & -0.8552 \\ -0.8552 & 1.8093 \end{bmatrix}, \\
\boldsymbol{R}_{h11} =& \begin{bmatrix} 0.0059 & -0.0020 \\ -0.0020 & 0.0080 \end{bmatrix}, \; \boldsymbol{R}_{v11} = \begin{bmatrix} 0.0065 & -0.0019 \\ -0.0019 & 0.0062 \end{bmatrix}
\end{aligned}$$

with $\epsilon = 800$ *and hence the following controller matrices are computed*

$$\boldsymbol{A}_c = \begin{bmatrix} -0.1783 & -0.8035 & 0.1885 & 0.5879 \\ 0.1338 & 0.4647 & -0.2754 & -0.7546 \\ 0.2620 & 0.0693 & -0.4416 & -1.2719 \\ -0.1745 & -0.2498 & -0.0896 & -0.2251 \end{bmatrix}, \; \boldsymbol{B}_c = \begin{bmatrix} -29.8481 & 2.9873 \\ -16.4800 & -3.5272 \\ -122.5966 & 56.0848 \\ -36.9975 & -89.7781 \end{bmatrix},$$

$$\boldsymbol{C}_c = \begin{bmatrix} 0.0001 & 0.0007 & 0.0011 & 0.0032 \\ 0.0007 & 0.0009 & -0.0005 & -0.0014 \end{bmatrix}$$

Consequently, the guaranteed cost of uncertain closed-loop process satisfies $J < 1.6515 \cdot 10^6$.

5.8. Concluding remarks

The purpose of this chapter was to demonstrate the use of LMI methods to obtain new results on the design of control laws for differential and discrete LRPs. The first part develops an $\mathcal{H}_\infty$ setting for the design of a static control law which, noting the physical basis of these processes in particular, their links to ILC mean that it is much more powerful than for 2-D linear systems. These results have then been extended to the cases when there are uncertainties in the process models. We also show that all these results can be extended to the use of a dynamic controller actuated by the previous pass profile which, by the process structure, is available for use.

In the second part of this chapter a guaranteed cost control problems have been formulated and solved using LMI methods. These are the first major results on control for performance for such processes and again the cost function used is well grounded in terms of the process dynamics and the requirements of industrial examples.

Chapter 6

LMI METHODS FOR 2-D SYSTEMS WITH STATE DELAYS

It is clear from many practical examples (Górecki *et al.*, 1989; Kolmanovskii and Myshkis, 1999; Malek-Zavarei and Jamshidi, 1987; Niculescu, 2001) that time-delay systems constitute the important class of control systems. Therefore, analysis and synthesis of time-delay systems have been made by many researches and, in turn, many results have been presented. Unfortunately, the main focus is on 1-D systems and there is no result in the area of 2-D(n-D) systems with state delays. Indeed, there is a large number of applications for 2-D(n-D) systems where state delays are unavoidable and must be take into account, e.g. during computation in 2-D framework, but to date these systems have not been considered.

According to the lack of results on analysis and synthesis of 2-D systems with state delays, this chapter provides them. It is shown that analysis and synthesis of 2-D systems with delays become possible because an efficient computer software based on LMI methods is available.

Here, Lyapunov-like techniques based on Lyapunov-Krasovski functionals (Boukas and Liu, 2003; Kharitonov and Zhabko, 2003; Mahmoud, 2000) have been utilized to derive sufficient conditions for the stability of 2-D systems with delays in terms of LMI. This approach gives an effective and implementable way to deal with uncertainties because other approaches, e.g. through augmentation of the local state vectors (Mahmoud, 2000; Xu *et al.*, 2001; Young, 2001), usually add undue complications to an uncertainty structure.

The chapter is organized as follows: First we provide the conditions for 2-D systems in both single and multiple delay cases. Some connections between delay systems and n-D delay-free systems will be established. Based on the derived results, it will be shown how to obtain conditions for stability and stabilisation under norm-bounded uncertainties.

6.1. Stability and stabilisation of 2-D system with state delays

The class of 2-D systems with delays under consideration, is represented by FMM with state delays (see-(2.28)) of the form

$$\begin{aligned} x(i+1,j+1) =& \boldsymbol{A}_1 x(i+1,j) + \boldsymbol{A}_2 x(i,j+1) + \boldsymbol{A}_{1d} x(i+1,j-d_1) \\ &+ \boldsymbol{A}_{2d} x(i-d_2,j+1) + \boldsymbol{B}_1 u(i+1,j) + \boldsymbol{B}_2 u(i,j+1) \\ y(i,j) =& \boldsymbol{C} x(i,j) + \boldsymbol{D} u(i,j) \end{aligned} \tag{6.1}$$

where $x(i,j) \in \mathbb{R}^n$ is the local state vector, $u(i,j) \in \mathbb{R}^l$ is the input vector, $y(i,j) \in \mathbb{R}^m$ is the output vector and d_1, d_2 are constant positive scalars representing delays along the vertical direction and horizontal direction respectively. The boundary conditions are given by

$$\begin{aligned} X_h(d_2) =& \{x(i,j) \quad \forall j \geq 0;\ i = -d_2, -d_2+1, \dots, 0\} \\ X_v(d_1) =& \{x(i,j) \quad \forall i \geq 0;\ j = -d_1, -d_1+1, \dots, 0\} \end{aligned} \tag{6.2}$$

For our purposes, denote $X_r = \sup\{\|x(i,j)\| : i+j = r,\ \ i,j \in \mathbb{Z}\}$, which allows us to define the asymptotic stability of the model (6.1).

Definition 6.1. *The 2-D linear state-delayed system (6.1) is said to be asymptotically stable if $\lim\limits_{r\to\infty} X_r = 0$ for zero input $u(i,j) = 0$ and any bounded boundary conditions of (6.2).*

The following theorem gives us a sufficient condition for system (6.1) to be asymptotically stable for any $d_1 \in [0,\infty)$ and $d_2 \in [0,\infty)$ (delay independent stability).

Theorem 6.1. *The 2-D state-delayed system (6.1) is asymptotically stable if there exist matrices $\boldsymbol{P} \succ 0$, $\boldsymbol{Q} \succ 0$, $\boldsymbol{Q}_1 \succ 0$, $\boldsymbol{Q}_2 \succ 0$ such that the following LMI holds*

$$\begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \\ \boldsymbol{A}_{1d}^T \\ \boldsymbol{A}_{2d}^T \end{bmatrix} \boldsymbol{P} \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_{1d} & \boldsymbol{A}_{2d} \end{bmatrix} - \begin{bmatrix} \boldsymbol{P}-\boldsymbol{Q}-\boldsymbol{Q}_1-\boldsymbol{Q}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{Q}_2 \end{bmatrix} \prec 0 \tag{6.3}$$

Proof. *Let a function $V(\zeta,\xi)$ that expresses the energy stored in the point $x(i+\zeta, j+\xi)$ be defined as*

$$V(\zeta,\xi) = x^T(i+\zeta, j+\xi) \boldsymbol{W}_{\zeta\xi} x(i+\zeta, j+\xi) \tag{6.4}$$

where $\boldsymbol{W}_{\zeta\xi} \succ 0$ is given and $\zeta \geq -d_1, \xi \geq -d_2$. To utilise the Lyapunov-Krasovskii approach in establishing the result, we introduce the Lyapunov function candidates for the delayed terms as

$$\begin{aligned} V_{d_1}(\zeta,\xi) &= V(\zeta,\xi) + \sum_{\theta=-d_1}^{-1} x^T(i+\zeta, j+\theta) \boldsymbol{U}_{\zeta\xi} x(i+\zeta, j+\theta) \\ V_{d_2}(\zeta,\xi) &= V(\zeta,\xi) + \sum_{\theta=-d_2}^{-1} x^T(i+\theta, j+\xi) \boldsymbol{U}_{\zeta\xi} x(i+\theta, j+\xi) \end{aligned} \tag{6.5}$$

where $\boldsymbol{U}_{\zeta\xi} \succ 0$ is given. In order to represent the change of the energy in both sides of (2.28), consider the increment $\Delta V(i,j)$ given by

$$\Delta V(i,j) = V(1,1) - V_{d_1}(1,0) - V_{d_2}(0,1) \tag{6.6}$$

Substituting (6.4) and (6.5) into (6.6), we obtain

$$\begin{aligned}\Delta V(i,j) = {} & [\boldsymbol{A}_1 x(i{+}1,j) + \boldsymbol{A}_2 x(i,j{+}1) + \boldsymbol{A}_{1d} x(i{+}1,j{-}d_1) + \boldsymbol{A}_{2d} x(i{-}d_2,j{+}1)]^T \\ & \times \boldsymbol{W}_{11}[\boldsymbol{A}_1 x(i{+}1,j) {+} \boldsymbol{A}_2 x(i,j{+}1) {+} \boldsymbol{A}_{1d} x(i{+}1,j{-}d_1) + \boldsymbol{A}_{2d} x(i{-}d_2,j{+}1)] \\ & - x^T(i{+}1,j)\boldsymbol{W}_{10} x(i{+}1,j) - \sum_{\theta=-d_1}^{-1} x^T(i{+}1,j{+}\theta)\boldsymbol{U}_{10} x(i{+}1,j{+}\theta) \\ & - x^T(i,j{+}1)\boldsymbol{W}_{01} x(i,j{+}1) - \sum_{\theta=-d_2}^{-1} x^T(i{+}\theta,j{+}1)\boldsymbol{U}_{01} x(i{+}\theta,j{+}1)\end{aligned}$$

After arranging the terms in the above equation, we have

$$\Delta V(i,j) = \widehat{x}^T \boldsymbol{\Pi} \widehat{x} \tag{6.7}$$

where

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \\ \boldsymbol{A}_{1d}^T \\ \boldsymbol{A}_{2d}^T \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \boldsymbol{W}_{11} \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_{1d} & \boldsymbol{A}_{2d} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{W}_{10} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{W}_{01} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{U}_{10} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{U}_{01} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{\Omega}_{10} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{\Omega}_{01} \end{bmatrix},$$

$$\begin{aligned}\widehat{x}^T &= \big[\, x^T(i{+}1,j)\; x^T(i,j{+}1)\; x^T(i{+}1,j{-}d_1)\; x^T(i{-}d_2,j{+}1)\; x^T(i{+}1,j{-}1) \\ &\qquad \cdots x^T(i{+}1,j{-}d_1{+}1)\; x^T(i{-}1,j{+}1) \cdots x^T(i{-}d_2{+}1,j{+}1) \big], \\ \boldsymbol{\Omega}_{10} &= \mathrm{diag}\left(\boldsymbol{U}_{10}, \quad \boldsymbol{U}_{10}, \quad \ldots, \quad \boldsymbol{U}_{10}\right), \quad (d_1 \textit{ terms}) \\ \boldsymbol{\Omega}_{01} &= \mathrm{diag}\left(\boldsymbol{U}_{01}, \quad \boldsymbol{U}_{01}, \quad \ldots, \quad \boldsymbol{U}_{01}\right), \quad (d_2 \textit{ terms})\end{aligned}$$

In the case when $\Delta V(i,j) < 0$ for $\widehat{x} \neq 0$, then a 2-D discrete linear system is asymptotically stable. In order to guarantee this stability condition it is clear that $\boldsymbol{\Pi} \prec 0$ has to hold. The last two rows and columns (blocks that only consist of $-\boldsymbol{\Omega}_{10}$ and $-\boldsymbol{\Omega}_{01}$) in (6.7) can be omitted because these terms are always negative definite. Thus we immediately obtain

$$\begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \\ \boldsymbol{A}_{1d}^T \\ \boldsymbol{A}_{2d}^T \end{bmatrix} \boldsymbol{W}_{11} \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_{1d} & \boldsymbol{A}_{2d} \end{bmatrix} - \begin{bmatrix} \boldsymbol{W}_{10} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{W}_{01} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{U}_{10} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{U}_{01} \end{bmatrix} \prec 0$$

Based on (6.4), the above inequality can be rewritten effectively as

$$V(1,1) < (V(1,0) + V(1,-d_1)) + (V(0,1) + V(-d_2,1)) \tag{6.8}$$

This means that the energy associated with the quadratic form $V_(1,1)$ *at the point* $(1,1)$ *on* $i+j=2$ *is strictly less than those at the adjacent points* $(1,0)$ *and* $(0,1)$ *and their delayed points* $(1,-d_1)$ *and* $(-d_2,1)$. *Notice that* $(1,0)$ *and* $(0,1)$ *are located on* $i+j=1$. *Thus, the condition (6.8) guarantees a local dissipative property. By considering the same effect at points along the lines* $i+j=K+1$ *and* $i+j=K$ *(see Lemma 2) for any nonnegative integer* K *and taking into account the energy transferred to a point at* $i+j=K$ *is passed to two points along the line* $i+j=K+1$ *along two directions (horizontal and vertical) and to one point along the line* $i+j=K+1+d_1$ *(vertical) and* $i+j=K+1+d_2$ *(horizontal) respectively, we choose*

$$\boldsymbol{W}_{11}=\boldsymbol{P},\quad \boldsymbol{W}_{10}=\boldsymbol{P}-\boldsymbol{Q}-\boldsymbol{Q}_1-\boldsymbol{Q}_2,\quad \boldsymbol{W}_{01}=\boldsymbol{Q},\quad \boldsymbol{U}_{10}=\boldsymbol{Q}_1,\quad \boldsymbol{U}_{01}=\boldsymbol{Q}_2$$

The condition (6.8) further implies that

$$\sum_{\zeta+\xi=K+1} V(\zeta,\xi) \le \sum_{\zeta+\xi=K} V(\zeta,\xi)$$

where the equality sign holds only when

$$\sum_{\zeta+\xi=K} V(\zeta,\xi)=0$$

Consequently, from (Hinamoto, 1989), we have $\lim\limits_{\zeta+\xi\to\infty}\|x(i+\zeta,j+\xi)\|=0$. *The asymptotic stability of the system is established and the proof is complete.* ■

It is straightforward to see that the LMI condition (6.3) is simple in computer implementation and effective because the number of decision variables to be computed is $2n(n+1)$, where n is the dimension of the local state vector ($x(i,j)\in\mathbb{R}^n$).

Example 6.1. *Consider the following 2-D state-delayed system of type (6.1)*

$$\boldsymbol{A}_1=\begin{bmatrix}0.1 & 0.4\\ 0.3 & 0.2\end{bmatrix},\ \boldsymbol{A}_2=\begin{bmatrix}0.1 & -0.3\\ 0.3 & 0\end{bmatrix},\ \boldsymbol{A}_{1d}=\begin{bmatrix}0.1 & -0.3\\ 0.3 & 0\end{bmatrix},\ \boldsymbol{A}_{2d}=\begin{bmatrix}0.1 & 0\\ 0.1 & -0.2\end{bmatrix} \quad (6.9)$$

In this case, LMI (6.3) is feasible and the matrices are

$$\boldsymbol{P}=\begin{bmatrix}42.6997 & -8.8064\\ -8.8064 & 49.2817\end{bmatrix},\quad \boldsymbol{Q}=\begin{bmatrix}16.4171 & -7.6379\\ -7.6379 & 14.6225\end{bmatrix},$$
$$\boldsymbol{Q}_1=\begin{bmatrix}7.0795 & 1.2712\\ 1.2712 & 5.9217\end{bmatrix},\quad \boldsymbol{Q}_2=\begin{bmatrix}7.2570 & -6.0167\\ -6.0167 & 10.6735\end{bmatrix}$$

This means that the system (6.9) is asymptotically stable independent of the delay sizes according to Theorem 6.1.

6.1.1. Connection between 2-D delay-free systems and 1-D state-delayed systems

The interesting point to note is that existing computer procedures for stability checking 2-D(n-D) systems can be immediately used to analyse 1-D systems with state delays. To see this, notice that Theorem 6.1 generalised the results for 2-D delay-free systems and 1-D state-delayed systems. Specifically, we consider the 2-D delay-free system

$$x(i+1,j+1) = \boldsymbol{A}_1 x(i+1,j) + \boldsymbol{A}_2 x(i,j+1) \tag{6.10}$$

and the 1-D state-delayed system

$$x(k+1) = \boldsymbol{A}_1 x(k) + \boldsymbol{A}_{1d} x(k-d) \tag{6.11}$$

By deleting appropriate rows and columns in (6.3) and considering the redundancy of certain positive definite variable matrices, we obtain the following corollaries.

Corollary 6.1. *The 2-D delay-free system (6.10) is asymptotically stable if there exist matrices $\boldsymbol{P} \succ 0$ and $\boldsymbol{Q} \succ 0$ such that the following LMI holds:*

$$\begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \end{bmatrix} \boldsymbol{P} \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{P}-\boldsymbol{Q} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q} \end{bmatrix} \prec 0 \tag{6.12}$$

Corollary 6.1 recovers the asymptotic stability result of 2-D delay-free system given by Lemma 1. It implies that the characteristic polynomial (2.8) has no zeros inside closed unit bidisc $\overline{U}^2 = \{(z_1,z_2) : |z_1| \leq 1, |z_2| \leq 1\}$.

Corollary 6.2. *The 1-D state-delayed system (6.11) is asymptotically stable if there exist matrices $\boldsymbol{P} \succ 0$ and $\boldsymbol{Q}_1 \succ 0$ such that the following LMI holds:*

$$\begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_{1d}^T \end{bmatrix} \boldsymbol{P} \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_{1d} \end{bmatrix} - \begin{bmatrix} \boldsymbol{P}-\boldsymbol{Q}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_1 \end{bmatrix} \prec 0 \tag{6.13}$$

By comparing the two corollaries, it can be observed that the LMIs in (6.12) and (6.13) are having an identical LMI structure which provides asymptotic stability for the two types of systems. In other words, when $\boldsymbol{A}_{1d}$ in the 1-D time-delay system is identified with $\boldsymbol{A}_2$ in the 2-D delay-free system, the delayed signal in the 1-D case can be viewed as a signal transmitting through another dimension in the 2-D framework.

Remark 6.1. *The asymptotic stability of the 2-D delay-free system (6.10) is equivalent to having no (z_1,z_2) in the unit bidisc such that $\det(\boldsymbol{I} - z_1\boldsymbol{A}_1 - z_2\boldsymbol{A}_2) = 0$. This clearly implies that there is no z in the unit disc such that $\det(\boldsymbol{I} - z\boldsymbol{A}_1 - z^{d+1}\boldsymbol{A}_2) = 0$ for any nonnegative integer d which corresponds to the asymptotic stability of the 1-D state-delayed system (6.11).*

Thus, it can be seen that the asymptotic stability of 2-D delay-free systems is a fairly strong condition imposing on the system matrices as compared to the *asymptotic stability of 1-D State-delayed systems. The connection can be extended to between n-D delay-free systems and 1-D systems with n different time delays. Other interesting observations related to delay differential equations and to 2-D polynomials can be found in (Agathoklis and Foda, 1989; Chiasson et al., 1985; Loiseau and Brethé, 1997).*

6.1.2. Multiple state-delayed case

Consider the 2-D multiple state-delayed system represented by

$$\begin{aligned} x(i+1,j+1) =&\boldsymbol{A}_1 x(i{+}1,j) + \boldsymbol{A}_2 x(i,j{+}1) + \sum_{k=1}^{s_1} \boldsymbol{A}_{1kd} x(i+1, j-d_{1k}) \\ &+ \sum_{l=1}^{s_2} \boldsymbol{A}_{2ld} x(i{-}d_{2l}, j{+}1) + \boldsymbol{B}_1 u(i{+}1,j) + \boldsymbol{B}_2 u(i,j{+}1) \end{aligned} \tag{6.14}$$

where $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_{1kd}$, $k = 1,\dots,s_1$, $\boldsymbol{A}_{2ld}$, $l = 1,\dots,s_2$, and $\boldsymbol{B}_1$, $\boldsymbol{B}_2$ are known constant matrices, s_1, s_2 denote the number of delayed terms in each direction. Additionally we assume that $d_{11} < d_{12} < \cdots < d_{1s_1}$ and $d_{21} < d_{22} < \cdots < d_{2s_2}$. In this case the boundary conditions are defined as

$$\begin{aligned} X_h(d_{2s_2}) =&\{x(i,j) \quad \forall j \geq 0; i = -d_{2s_2}, -d_{2s_2}+1, \dots, 0\} \\ X_v(d_{1s_1}) =&\{x(i,j) \quad \forall i \geq 0; j = -d_{1s_1}, -d_{1s_1}+1, \dots, 0\} \end{aligned} \tag{6.15}$$

Theorem 6.2. *The 2-D multiple state-delayed system (6.14) is asymptotically stable if there exist matrices* $\boldsymbol{P} \succ 0$, $\boldsymbol{Q} \succ 0$, $\boldsymbol{U}_{11},\dots,\boldsymbol{U}_{1s_1} \succ 0$ *and* $\boldsymbol{U}_{21},\dots,\boldsymbol{U}_{2s_2} \succ 0$ *such that the following LMI holds*

$$\begin{bmatrix} \boldsymbol{A}_1^T \\ \boldsymbol{A}_2^T \\ \boldsymbol{\Lambda}_{1d}^T \\ \boldsymbol{\Lambda}_{2d}^T \end{bmatrix} \boldsymbol{P} \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{\Lambda}_{1d} & \boldsymbol{\Lambda}_{2d} \end{bmatrix} - \begin{bmatrix} \boldsymbol{P}-\boldsymbol{Q}-\boldsymbol{\Phi}_1-\boldsymbol{\Phi}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Omega}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Omega}_2 \end{bmatrix} \prec 0 \tag{6.16}$$

where

$$\begin{gathered} \boldsymbol{\Lambda}_{1d} = \begin{bmatrix} \boldsymbol{A}_{11d}, \, \boldsymbol{A}_{12d}, \, \dots, \, \boldsymbol{A}_{1s_1d} \end{bmatrix}, \; \boldsymbol{\Lambda}_{2d} = \begin{bmatrix} \boldsymbol{A}_{21d}, \, \boldsymbol{A}_{22d}, \, \dots, \, \boldsymbol{A}_{2s_2d} \end{bmatrix}, \\ \boldsymbol{\Omega}_1 = \operatorname{diag}\left(\boldsymbol{Q}_{11}, \, \boldsymbol{Q}_{12}, \, \dots, \, \boldsymbol{Q}_{1s_1}\right), \; \boldsymbol{\Omega}_2 = \operatorname{diag}\left(\boldsymbol{Q}_{21}, \quad \boldsymbol{Q}_{22}, \quad \dots, \quad \boldsymbol{Q}_{2s_2}\right), \\ \boldsymbol{Q}_{1k} = \sum_{\theta=1}^{s_1-k+1} \boldsymbol{U}_{1\theta}, \; \boldsymbol{Q}_{2l} = \sum_{\theta=1}^{s_2-l+1} \boldsymbol{U}_{2\theta}, \; \boldsymbol{\Phi}_1 = \sum_{k=1}^{s_1} \boldsymbol{Q}_{1k}, \quad \boldsymbol{\Phi}_2 = \sum_{l=1}^{s_2} \boldsymbol{Q}_{2l} \end{gathered}$$

Proof. *It can be established in the same manner as in Theorem 6.1 with*

$$\begin{aligned} V_{d_1}(\zeta,\xi) =& x^T(i+\zeta,j+\xi)\boldsymbol{W}_{\zeta\xi}x(i+\zeta,j+\xi) \\ &+\sum_{k=1}^{s1}\sum_{\theta=-d_{1k}}^{-1} x^T(i+\zeta,j+\theta)\boldsymbol{U}_{1k}x(i+\zeta,j+\theta) \\ V_{d_2}(\zeta,\xi) =& x^T(i+\zeta,j+\xi)\boldsymbol{W}_{\zeta\xi}x(i+\zeta,j+\xi) \\ &+\sum_{l=1}^{s_2}\sum_{\theta=-d_{2l}}^{-1} x^T(i+\theta,j+\xi)\boldsymbol{U}_{2l}x(i+\theta,j+\xi) \end{aligned}$$

The LMI then follows according to the choice of matrices as

$$\boldsymbol{W}_{11}=\boldsymbol{P},\ \boldsymbol{Q}_{1k}=\sum_{\theta=1}^{s_1-k+1}\boldsymbol{U}_{1\theta},\ \boldsymbol{Q}_{2l}=\sum_{\theta=1}^{s_2-l+1}\boldsymbol{U}_{2\theta},\ \boldsymbol{W}_{01}=\boldsymbol{Q},$$
$$\boldsymbol{W}_{10}=\boldsymbol{P}-\boldsymbol{Q}-\boldsymbol{Q}_{11}-\ldots-\boldsymbol{Q}_{1k}-\boldsymbol{Q}_{21}-\ldots-\boldsymbol{Q}_{2l}$$

■

6.1.3. Commensurate delays case

One potential problem with the computing of the condition (6.16), however, is the fact that the dimensions of the matrices involved in the LMI based conditions could well be very large and hence numerical difficulties could arise. This can occur, for example, when the system dimensionality is large ($n \gg 1$) and/or many delays are present.

It turns out that in case of commensurate delays in Theorem 6.2 the number of decision variables can be reduced. To proceed, the following definition is required

Definition 6.2. *(Niculescu, 2001) Delays $h_1, \cdots, h_q$ are termed noncommensurate if $\exists$ no integers $l_1, \cdots, l_q$ (not all of them zero) such that $\sum_{i=1}^{q} l_i h_i = 0$. The underlying delay differential system is termed commensurate if $q = 1$.*

It is shown that if all delays present in (6.16) are commensurate, then investigation of the stability properties of a 2-D delay system can be treated equivalently as the stability investigation of a 4-D delay free system. The key to establishing this fact is the Elementary Operation Algorithm (EOA) developed by Gałkowski (Gałkowski, 2001a). The basic idea behind this algorithm is the subsequent use of elementary operations over multivariable polynomials and the matrix size augmentation which preserves the matrix determinant, to obtain the required state-space realization. To see the application of EOA, consider the following example.

Example 6.2. *In general case, the notation associated with this area is very cumbersome and therefore ease of presentation we only consider the particular* *case of a 2-D linear system of the form (6.14) with two delays in each direction, i.e. we restrict attention to* $m_1 = m_2 = 2$. *In which case it is clear that the associated characteristic polynomial for stability is given by the determinant of the following 2-D polynomial matrix*

$$\boldsymbol{I} - \boldsymbol{A}_1 z_1^{-1} - \boldsymbol{A}_2 z_2^{-1} - \boldsymbol{A}_3 z_1^{-h_1 k} - \boldsymbol{A}_4 z_1^{-h_2 k} - \boldsymbol{A}_5 z_2^{-p_1 l} - \boldsymbol{A}_6 z_1^{-p_2 l} \qquad (6.17)$$

where real scalars k,l *are positive and* $h_1,\ h_2,\ p_1,\ p_2$ *are natural numbers. Now introduce the new variables* $z_1^k = z_3,\ z_2^l = z_4$ *and then rewrite (6.17) as*

$$\boldsymbol{I} - \boldsymbol{A}_1 z_1^{-1} - \boldsymbol{A}_2 z_2^{-1} - \boldsymbol{A}_3 z_3^{-h_1} - \boldsymbol{A}_4 z_3^{-h_2} - \boldsymbol{A}_5 z_4^{-p_1} - \boldsymbol{A}_6 z_4^{-p_2}$$

Assume also that $h_1 = 1,\ h_2 = 2,\ p_1 = 1,\ p_2 = 2$ *which yields*

$$\boldsymbol{I} - \boldsymbol{A}_1 z_1^{-1} - \boldsymbol{A}_2 z_2^{-1} - \boldsymbol{A}_3 z_3^{-1} - \boldsymbol{A}_4 z_3^{-2} - \boldsymbol{A}_5 z_4^{-1} - \boldsymbol{A}_6 z_4^{-2} \qquad (6.18)$$

Application of the EOA to this last 4-D polynomial matrix now gives

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & z_4^{-1}\boldsymbol{A}_5 \\ \boldsymbol{0} & \boldsymbol{I} & z_3^{-1}\boldsymbol{A}_4 \\ z_4^{-1}\boldsymbol{I} & z_3^{-1}\boldsymbol{I} & \boldsymbol{I} - \boldsymbol{A}_1 z_1^{-1} - \boldsymbol{A}_2 z_2^{-1} - \boldsymbol{A}_3 z_3^{-1} - \boldsymbol{A}_6 z_4^{-1} \end{bmatrix} \qquad (6.19)$$

which is equivalent to

$$\boldsymbol{I} - \widehat{\boldsymbol{A}}_1 z_1^{-1} - \widehat{\boldsymbol{A}}_2 z_2^{-1} - \widehat{\boldsymbol{A}}_3 z_3^{-1} - \widehat{\boldsymbol{A}}_4 z_4^{-1} \qquad (6.20)$$

where

$$\widehat{\boldsymbol{A}}_1 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_1 \end{bmatrix}, \widehat{\boldsymbol{A}}_2 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_2 \end{bmatrix}, \widehat{\boldsymbol{A}}_3 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{A}_4 \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{A}_3 \end{bmatrix}, \widehat{\boldsymbol{A}}_4 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{A}_5 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{A}_6 \end{bmatrix} \qquad (6.21)$$

Here only elementary operations that preserve the matrix determinant are used, hence it is straightforward to see that (6.18) and (6.20) have the same determinant and the stability property for both system descriptions is the same. Indeed, the system with the characteristic polynomial represented by (6.17) is stable if there exist $\boldsymbol{P} \succ 0,\ \boldsymbol{Q} \succ 0,\ \boldsymbol{Q}_1 \succ 0,\ \boldsymbol{Q}_2 \succ 0$ *such that the following LMI holds*

$$\begin{bmatrix} \widehat{\boldsymbol{A}}_1^T \\ \widehat{\boldsymbol{A}}_2^T \\ \widehat{\boldsymbol{A}}_3^T \\ \widehat{\boldsymbol{A}}_4^T \end{bmatrix} \boldsymbol{P} \begin{bmatrix} \widehat{\boldsymbol{A}}_1 & \widehat{\boldsymbol{A}}_2 & \widehat{\boldsymbol{A}}_3 & \widehat{\boldsymbol{A}}_4 \end{bmatrix} - \begin{bmatrix} \boldsymbol{P} - \boldsymbol{Q} - \boldsymbol{Q}_1 - \boldsymbol{Q}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_2 \end{bmatrix} \prec 0$$

where the matrices $\widehat{\boldsymbol{A}}_1,\ \widehat{\boldsymbol{A}}_2,\ \widehat{\boldsymbol{A}}_3,\ \widehat{\boldsymbol{A}}_4$ *are defined in (6.21).*

It is clear that the above result is easily extended to the partially commensurate case. In particular, assume that each delay $d_{1v},\ v = 1, \ldots, m_1$ is a multiple of one of the basic noncommensurate delays $k_1, k_2, \ldots, k_{t_1}$ and similarly for $d_{1h},\ h = 1, \ldots, m_2$ of $l_1, l_2, \ldots, l_{t_2}$. Then the previous method exploiting this fact requires the investigation of an n-D linear system, where $n = t_1 + t_2 + 2$ whereas the method of Theorem 1 here requires the investigation of an m-D linear system with $m = m_1 + m_2 + 2$.

6.1.4. Stabilisation of 2-D systems with delays

Consider the 2-D state-delayed system (2.28) and assume that the following state feedback control law is used

$$u(i,j) = \boldsymbol{K}x(i,j) \tag{6.22}$$

The corresponding closed-loop system is

$$\begin{aligned} x(i+1,j+1) = {} & (\boldsymbol{A}_1 + \boldsymbol{B}_1\boldsymbol{K})\, x(i+1,j) + (\boldsymbol{A}_2 + \boldsymbol{B}_2\boldsymbol{K})\, x(i,j+1) \\ & + \boldsymbol{A}_{1d}x(i+1,j-d_1) + \boldsymbol{A}_{2d}x(i-d_2,j+1) \end{aligned} \tag{6.23}$$

If there $\boldsymbol{K}$ exists so that (6.23) is asymptotically stable, then the 2-D state-delayed system (2.28) is said to be stabilisable.

Theorem 6.3. *The 2-D state-delayed system (2.28) is stabilisable with control law (6.22) if there exist matrices* $\boldsymbol{W} \succ 0$*,* $\boldsymbol{Z} \succ 0$*,* $\boldsymbol{Z}_1 \succ 0$*,* $\boldsymbol{Z}_2 \succ 0$ *and* $\boldsymbol{N}$ *such that*

$$\begin{bmatrix} -\boldsymbol{W} & \boldsymbol{A}_1\boldsymbol{W}+\boldsymbol{B}_1\boldsymbol{N} & \boldsymbol{A}_2\boldsymbol{W}+\boldsymbol{B}_2\boldsymbol{N} & \boldsymbol{A}_{1d}\boldsymbol{W} & \boldsymbol{A}_{2d}\boldsymbol{W} \\ \boldsymbol{W}\boldsymbol{A}_1^T+\boldsymbol{N}^T\boldsymbol{B}_1^T & \boldsymbol{W}-\boldsymbol{Z}-\boldsymbol{Z}_1-\boldsymbol{Z}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_2^T+\boldsymbol{N}^T\boldsymbol{B}_2^T & \boldsymbol{0} & -\boldsymbol{Z} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_{1d}^T & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Z}_1 & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_{2d}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Z}_2 \end{bmatrix} \prec 0 \tag{6.24}$$

In this case, a stabilising matrix $\boldsymbol{K}$ *is given by* $\boldsymbol{N}\boldsymbol{W}^{-1}$*.*

Proof. *Based on (6.3) and (6.22), the closed-loop system is asymptotically stable if there exist* $\boldsymbol{P} \succ 0$*,* $\boldsymbol{Q} \succ 0$*,* $\boldsymbol{Q}_1 \succ 0$*,* $\boldsymbol{Q}_2 \succ 0$*, such that*

$$\begin{bmatrix} (\boldsymbol{A}_1+\boldsymbol{B}_1\boldsymbol{K})^T \\ (\boldsymbol{A}_2+\boldsymbol{B}_2\boldsymbol{K})^T \\ \boldsymbol{A}_{1d}^T \\ \boldsymbol{A}_{2d}^T \end{bmatrix} \boldsymbol{P} \begin{bmatrix} \boldsymbol{A}_1+\boldsymbol{B}_1\boldsymbol{K} & \boldsymbol{A}_2+\boldsymbol{B}_2\boldsymbol{K} & \boldsymbol{A}_{1d} & \boldsymbol{A}_{2d} \end{bmatrix} - \begin{bmatrix} \boldsymbol{P}-\boldsymbol{Q}-\boldsymbol{Q}_1-\boldsymbol{Q}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_2 \end{bmatrix} \prec 0$$

Applying the Schur complement formula to the above inequality followed by pre- and post-multiplying $\operatorname{diag}(\boldsymbol{P},\boldsymbol{I},\boldsymbol{I},\boldsymbol{I},\boldsymbol{I})$ *and its transpose, we obtain*

$$\begin{bmatrix} -\boldsymbol{P} & \boldsymbol{P}\boldsymbol{A}_1+\boldsymbol{P}\boldsymbol{B}_1\boldsymbol{K} & \boldsymbol{P}\boldsymbol{A}_2+\boldsymbol{P}\boldsymbol{B}_2\boldsymbol{K} & \boldsymbol{P}\boldsymbol{A}_{1d} & \boldsymbol{P}\boldsymbol{A}_{2d} \\ \boldsymbol{A}_1^T\boldsymbol{P}+\boldsymbol{K}^T\boldsymbol{B}_1^T\boldsymbol{P} & -\boldsymbol{P}+\boldsymbol{Q}+\boldsymbol{Q}_1+\boldsymbol{Q}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{A}_2^T\boldsymbol{P}+\boldsymbol{K}^T\boldsymbol{B}_2^T\boldsymbol{P} & \boldsymbol{0} & -\boldsymbol{Q} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{A}_{1d}^T\boldsymbol{P} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Q}_1 & \boldsymbol{0} \\ \boldsymbol{A}_{2d}^T\boldsymbol{P} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Q}_2 \end{bmatrix} \prec 0$$

Note that this last condition is bilinear in matrix variables $\boldsymbol{P}$ *and* $\boldsymbol{K}$ *and therefore it may be considered as a BMI problem (3.4), which is not amenable for effective computations (recall that BMI problems belong to the class of* $\mathcal{NP}$*-hard problems).*

However, this can be reformulated as an LMI problem and hence solved in polynomial time. To see this, let us define a new variable $\boldsymbol{W} = \boldsymbol{P}^{1}$ *and pre- and post-multiplying by* $\operatorname{diag}(\boldsymbol{W}, \boldsymbol{W}, \boldsymbol{W}, \boldsymbol{W}, \boldsymbol{W})$ *to yield*

$$\begin{bmatrix} -\boldsymbol{W} & \boldsymbol{A}_1\boldsymbol{W}+\boldsymbol{B}_1\boldsymbol{K}\boldsymbol{W} & \boldsymbol{A}_2\boldsymbol{W}+\boldsymbol{B}_2\boldsymbol{K}\boldsymbol{W} & \boldsymbol{A}_{1d}\boldsymbol{W} & \boldsymbol{A}_{2d}\boldsymbol{W} \\ \boldsymbol{W}\boldsymbol{A}_1^T+\boldsymbol{W}\boldsymbol{K}^T\boldsymbol{B}_1^T & \boldsymbol{\Upsilon} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_2^T+\boldsymbol{W}\boldsymbol{K}^T\boldsymbol{B}_2^T & \boldsymbol{0} & -\boldsymbol{W}\boldsymbol{Q}\boldsymbol{W} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_{1d}^T & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{W}\boldsymbol{Q}_1\boldsymbol{W} & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_{2d}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{W}\boldsymbol{Q}_2\boldsymbol{W} \end{bmatrix} \prec 0$$

where

$$\boldsymbol{\Upsilon} = -\boldsymbol{W}+\boldsymbol{W}\boldsymbol{Q}\boldsymbol{W}+\boldsymbol{W}\boldsymbol{Q}_1\boldsymbol{W}+\boldsymbol{W}\boldsymbol{Q}_2\boldsymbol{W}$$

and let $\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{Q}\boldsymbol{W}$, $\boldsymbol{Z}_1 = \boldsymbol{W}\boldsymbol{Q}_1\boldsymbol{W}$, $\boldsymbol{Z}_2 = \boldsymbol{W}\boldsymbol{Q}_2\boldsymbol{W}$, $\boldsymbol{N} = \boldsymbol{K}\boldsymbol{W}$ *we obtain the final form as in (6.24).* ■

6.2. Robust stability and robust stabilisation of 2-D systems with state delays

In this section, for brevity, we consider a 2-D uncertain system with single delays described by

$$\begin{aligned} x(i+1,j+1) =& (\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)x(i+1,j)+(\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)x(i,j+1) \\ &+(\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)x(i+1,j-d_1)+(\boldsymbol{A}_{1d}+\Delta\boldsymbol{A}_{2d})x(i-d_2,j+1) \\ &+(\boldsymbol{B}_1+\Delta\boldsymbol{B}_1)u(i+1,j)+(\boldsymbol{B}_2+\Delta\boldsymbol{B}_2)u(i,j+1) \end{aligned} \tag{6.25}$$

Suppose the uncertain matrix $\Delta\boldsymbol{A}$ be defined in the norm-bounded (Khargonekar *et al.*, 1990; Mahmoud, 2000) form as

$$\Delta\boldsymbol{A} = \begin{bmatrix} \Delta\boldsymbol{A}_1 & \Delta\boldsymbol{A}_2 & \Delta\boldsymbol{A}_{1d} & \Delta\boldsymbol{A}_{2d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}_1 & \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}_2 & \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}_{1d} & \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}_{2d} \end{bmatrix} \tag{6.26}$$

where $\boldsymbol{H}$, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_{1d}$, $\boldsymbol{E}_{2d}$ are known constant matrices with compatible dimensions and $\boldsymbol{\mathcal{F}}$ satisfies (4.5).

For further consideration we rewrite the uncertainty structure of (6.26) as

$$\Delta\boldsymbol{A} = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{H} & \boldsymbol{H} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{F}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{F}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_{1d} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_{2d} \end{bmatrix} \equiv \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{\mathcal{F}}}\widetilde{\boldsymbol{E}} \tag{6.27}$$

6.2.1. Robust stability

Theorem 6.4. *A 2-D state-delayed system (6.25) with uncertainty modelled by (4.5) and (6.27) is asymptotically stable if there exist matrices* $\boldsymbol{P} \succ 0$, $\boldsymbol{Q} \succ 0$,

$\boldsymbol{Q}_1 \succ 0$, $\boldsymbol{Q}_2 \succ 0$ *and a scalar* $\epsilon > 0$ *such that the following LMI holds*

$$\begin{bmatrix} -\boldsymbol{P} & \boldsymbol{P}\boldsymbol{A}_1 & \boldsymbol{P}\boldsymbol{A}_2 & \boldsymbol{P}\boldsymbol{A}_{1d} & \boldsymbol{P}\boldsymbol{A}_{2d} & \boldsymbol{P}\boldsymbol{H} \\ \boldsymbol{A}_1^T\boldsymbol{P} & \boldsymbol{\Upsilon} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_2^T\boldsymbol{P} & \mathbf{0} & -\boldsymbol{Q}+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_{1d}^T\boldsymbol{P} & \mathbf{0} & \mathbf{0} & -\boldsymbol{Q}_1+\epsilon\boldsymbol{E}_{1d}^T\boldsymbol{E}_{1d} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_{2d}^T\boldsymbol{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{Q}_2+\epsilon\boldsymbol{E}_{2d}^T\boldsymbol{E}_{2d} & \mathbf{0} \\ \boldsymbol{H}^T\boldsymbol{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -0.25\epsilon\boldsymbol{I} \end{bmatrix} \prec 0 \qquad (6.28)$$

where

$$\boldsymbol{\Upsilon} = -\boldsymbol{P}+\boldsymbol{Q}+\boldsymbol{Q}_1+\boldsymbol{Q}_2+\epsilon\boldsymbol{E}_1^T\boldsymbol{E}_1$$

Proof. *Based on Theorem 6.1, a sufficient condition for asymptotic stability of the 2-D state-delayed system (6.3) can be rewritten in the form to contain the uncertainty modelled by (4.5) and (6.27) as*

$$\left[\ \boldsymbol{A}^T + \Delta\boldsymbol{A}^T\ \right]\boldsymbol{P}\left[\ \boldsymbol{A} + \Delta\boldsymbol{A}\ \right] + \boldsymbol{S} \prec 0 \qquad (6.29)$$

where

$$\boldsymbol{A} = \left[\boldsymbol{A}_1\ \boldsymbol{A}_{1d}\ \boldsymbol{A}_2\ \boldsymbol{A}_{2d}\right],\ \boldsymbol{S} = \begin{bmatrix} -\boldsymbol{P}+\boldsymbol{Q}+\boldsymbol{Q}_1+\boldsymbol{Q}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{Q}_2 \end{bmatrix}$$

By applying the Schur complement to (6.29), we obtain

$$\begin{bmatrix} -\boldsymbol{P}^{-1} & \boldsymbol{A}+\Delta\boldsymbol{A} \\ \boldsymbol{A}^T+\Delta\boldsymbol{A}^T & \boldsymbol{S} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{P}^{-1} & \boldsymbol{A} \\ \boldsymbol{A}^T & \boldsymbol{S} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{F}}\widetilde{\boldsymbol{E}} \\ \widetilde{\boldsymbol{E}}^T\widetilde{\boldsymbol{F}}^T\widetilde{\boldsymbol{H}}^T & \mathbf{0} \end{bmatrix}$$

By using Lemma 8, (6.29) is implied by

$$\begin{bmatrix} -\boldsymbol{P}^{-1}+4\epsilon^{-1}\boldsymbol{H}\boldsymbol{H}^T & \boldsymbol{A}_1 & \boldsymbol{A}_{1d} & \boldsymbol{A}_2 & \boldsymbol{A}_{2d} \\ \boldsymbol{A}_1^T & \boldsymbol{\Upsilon} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_{1d}^T & \mathbf{0} & -\boldsymbol{Q}+\epsilon\boldsymbol{E}_{1d}^T\boldsymbol{E}_{1d} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_2^T & \mathbf{0} & \mathbf{0} & -\boldsymbol{Q}_1+\epsilon\boldsymbol{E}_2^T\boldsymbol{E}_2 & \mathbf{0} \\ \boldsymbol{A}_{2d}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{Q}_2+\epsilon\boldsymbol{E}_{2d}^T\boldsymbol{E}_{2d} \end{bmatrix} \prec 0$$

which is BMI form due to occurrence of the terms $\boldsymbol{P}$ *and* $\boldsymbol{P}^{-1}$. *However, by pre- and post-multiplying* $\operatorname{diag}(\boldsymbol{P}, \boldsymbol{I}, \boldsymbol{I}, \boldsymbol{I}, \boldsymbol{I})$ *and its transpose and considering the Schur complement, we obtain the LMI (6.28).* ■

Remark 6.2. *The extension to the multiple delay case follows in a similar way as in Theorem 6.2.*

6.2.2. Robust stabilisation

Consider the uncertain 2-D state-delayed system described by (6.25). The uncertainties associated with the state matrices are modelled by (4.5), (6.27) and additionally we have

$$\begin{bmatrix} \Delta\boldsymbol{B}_1 & \Delta\boldsymbol{B}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}_{1b} & \boldsymbol{H}\boldsymbol{\mathcal{F}}\boldsymbol{E}_{2b} \end{bmatrix} \tag{6.30}$$

With the same type of control law as (6.22), the corresponding closed-loop system is given by

$$\begin{aligned} x(i+1,j+1) = &((\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)+(\boldsymbol{B}_1+\Delta\boldsymbol{B}_1)\boldsymbol{K})\,x(i+1,j) \\ &+((\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)+(\boldsymbol{B}_2+\Delta\boldsymbol{B}_2)\boldsymbol{K})\,x(i,j+1) \\ &+(\boldsymbol{A}_1+\Delta\boldsymbol{A}_1)x(i+1,j-d_1)+(\boldsymbol{A}_{1d}+\Delta\boldsymbol{A}_{2d})x(i-d_2,j+1) \end{aligned} \tag{6.31}$$

A matrix $\boldsymbol{K}$ is said to be robustly stabilising if (6.31) is asymptotically stable for all uncertainties in (6.25) satisfying (4.5), (6.27) and (6.30). The system (6.25) is said to be robustly stabilisable with $\boldsymbol{K}$.

Theorem 6.5. *The 2-D uncertain state-delayed system (6.25) is robustly stabilisable with control law (6.22) if there exist matrices $\boldsymbol{W} \succ 0$, $\boldsymbol{Z} \succ 0$, $\boldsymbol{Z}_1 \succ 0$, $\boldsymbol{Z}_2 \succ 0$, $\boldsymbol{N}$ and a scalar $\epsilon > 0$ such that the following LMI holds:*

$$\begin{bmatrix} \boldsymbol{\Upsilon}_{11} & \boldsymbol{\Upsilon}_{12} \\ \boldsymbol{\Upsilon}_{12}^T & \boldsymbol{\Upsilon}_{22} \end{bmatrix} \prec 0 \tag{6.32}$$

where

$$\boldsymbol{\Upsilon}_{11} = \begin{bmatrix} -\boldsymbol{W}+4\epsilon\boldsymbol{H}\boldsymbol{H}^T & \boldsymbol{A}_1\boldsymbol{W}+\boldsymbol{B}_1\boldsymbol{N} & \boldsymbol{A}_2\boldsymbol{W}+\boldsymbol{B}_2\boldsymbol{N} & \boldsymbol{A}_{1d}\boldsymbol{W} & \boldsymbol{A}_{2d}\boldsymbol{W} \\ \boldsymbol{W}\boldsymbol{A}_1^T+\boldsymbol{N}^T\boldsymbol{B}_1^T & -\boldsymbol{W}+\boldsymbol{Z}+\boldsymbol{Z}_1+\boldsymbol{Z}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_2^T+\boldsymbol{N}^T\boldsymbol{B}_2^T & \boldsymbol{0} & -\boldsymbol{Z} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_{1d}^T & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Z}_1 & \boldsymbol{0} \\ \boldsymbol{W}\boldsymbol{A}_{2d}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{Z}_2 \end{bmatrix} \tag{6.33}$$

$$\boldsymbol{\Upsilon}_{12} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ (\boldsymbol{E}_1\boldsymbol{W}+\boldsymbol{E}_{1b}\boldsymbol{N})^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{E}_2\boldsymbol{W}+\boldsymbol{E}_{2b}\boldsymbol{N})^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}\boldsymbol{E}_{1d}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}\boldsymbol{E}_{2d}^T \end{bmatrix} \tag{6.34}$$

$$\boldsymbol{\Upsilon}_{22} = \operatorname{diag}\begin{pmatrix} -\epsilon\boldsymbol{I}, & -\epsilon\boldsymbol{I}, & -\epsilon\boldsymbol{I}, & -\epsilon\boldsymbol{I} \end{pmatrix} \tag{6.35}$$

In this case, a robustly stabilising matrix $\boldsymbol{K}$ is given by $\boldsymbol{N}\boldsymbol{W}^{-1}$.

Proof. *By incorporating the norm-bounded uncertainties to (6.24), we obtain*

$$\begin{bmatrix} -\boldsymbol{W} & \boldsymbol{A}_1\boldsymbol{W}+\boldsymbol{B}_1\boldsymbol{N} & \boldsymbol{A}_2\boldsymbol{W}+\boldsymbol{B}_2\boldsymbol{N} & \boldsymbol{A}_{1d}\boldsymbol{W} & \boldsymbol{A}_{2d}\boldsymbol{W} \\ \boldsymbol{W}\boldsymbol{A}_1^T+\boldsymbol{N}^T\boldsymbol{B}_1^T & -\boldsymbol{W}+\boldsymbol{Z}+\boldsymbol{Z}_1+\boldsymbol{Z}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}\boldsymbol{A}_2^T+\boldsymbol{N}^T\boldsymbol{B}_2^T & \mathbf{0} & -\boldsymbol{Z} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}\boldsymbol{A}_{1d}^T & \mathbf{0} & \mathbf{0} & -\boldsymbol{Z}_1 & \mathbf{0} \\ \boldsymbol{W}\boldsymbol{A}_{2d}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\boldsymbol{Z}_2 \end{bmatrix}$$
$$+\begin{bmatrix} \mathbf{0} & \Delta\boldsymbol{A}_1\boldsymbol{W}+\Delta\boldsymbol{B}_1\boldsymbol{N} & \Delta\boldsymbol{A}_2\boldsymbol{W}+\Delta\boldsymbol{B}_2\boldsymbol{N} & \Delta\boldsymbol{A}_{1d}\boldsymbol{W} & \Delta\boldsymbol{A}_{2d}\boldsymbol{W} \\ \boldsymbol{W}\Delta\boldsymbol{A}_1^T+\boldsymbol{N}^T\Delta\boldsymbol{B}_1^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}\Delta\boldsymbol{A}_2^T+\boldsymbol{N}^T\Delta\boldsymbol{B}_2^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}\Delta\boldsymbol{A}_{1d}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}\Delta\boldsymbol{A}_{2d}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \prec 0$$

The second term in the above inequality can be represented using the following expression

$$\begin{bmatrix} \boldsymbol{H} & \boldsymbol{H} & \boldsymbol{H} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_1\boldsymbol{W}+\boldsymbol{E}_{1b}\boldsymbol{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{E}_2\boldsymbol{W}+\boldsymbol{E}_{2b}\boldsymbol{N} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{E}_{1d}\boldsymbol{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_{2d}\boldsymbol{W} \end{bmatrix}$$

By carrying out the same operation as previously presented for (6.29), we can write

$$\begin{bmatrix} \mathbf{0} & \Delta\boldsymbol{A}_1\boldsymbol{W}+\Delta\boldsymbol{B}_1\boldsymbol{N} & \Delta\boldsymbol{A}_2\boldsymbol{W}+\Delta\boldsymbol{B}_2\boldsymbol{N} & \Delta\boldsymbol{A}_{1d}\boldsymbol{W} & \Delta\boldsymbol{A}_{2d}\boldsymbol{W} \\ \boldsymbol{W}\Delta\boldsymbol{A}_1^T+\boldsymbol{N}^T\Delta\boldsymbol{B}_1^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}\Delta\boldsymbol{A}_2^T+\boldsymbol{N}^T\Delta\boldsymbol{B}_2^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}\Delta\boldsymbol{A}_{1d}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{W}\Delta\boldsymbol{A}_{2d}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$
$$\preceq \mathrm{diag}\Big(4\epsilon\boldsymbol{H}\boldsymbol{H}^T,\ \epsilon^{-1}(\boldsymbol{E}_1\boldsymbol{W}+\boldsymbol{E}_{1b}\boldsymbol{N})^T(\boldsymbol{E}_1\boldsymbol{W}+\boldsymbol{E}_{1b}\boldsymbol{N}),$$
$$\epsilon^{-1}(\boldsymbol{E}_2\boldsymbol{W}+\boldsymbol{E}_{2b}\boldsymbol{N})^T(\boldsymbol{E}_2\boldsymbol{W}+\boldsymbol{E}_{2b}\boldsymbol{N}), \epsilon^{-1}\boldsymbol{W}\boldsymbol{E}_{1d}^T\boldsymbol{E}_{1d}\boldsymbol{W},\ \epsilon^{-1}\boldsymbol{W}\boldsymbol{E}_{2d}^T\boldsymbol{E}_{2d}\boldsymbol{W}\Big)$$

The result then follows by observing that

$$\boldsymbol{\Upsilon}_{11} - \boldsymbol{\Upsilon}_{12}\boldsymbol{\Upsilon}_{22}^{-1}\boldsymbol{\Upsilon}_{12}^T \prec 0$$

is a Schur complement of (6.32) where $\boldsymbol{\Upsilon}_{11}$, $\boldsymbol{\Upsilon}_{12}$, $\boldsymbol{\Upsilon}_{22}$ are described by (6.33)-(6.35). ■

Example 6.3. *To illustrate the results of Theorem 6.5, let us consider a system described by (6.25) with the following matrices*

$$\boldsymbol{A}_1 = \begin{bmatrix} -0.2450 & 0.0307 \\ -0.1444 & 0.0008 \end{bmatrix},\ \boldsymbol{A}_2 = \begin{bmatrix} 0.2860 & 0.1800 \\ -0.1435 & -0.4601 \end{bmatrix},\ \boldsymbol{A}_{1d} = \begin{bmatrix} 0.1453 & 0.1489 \\ 0.0824 & 0.0536 \end{bmatrix},$$
$$\boldsymbol{A}_{2d} = \begin{bmatrix} 0.0880 & 0.1367 \\ 0.1867 & 0.0425 \end{bmatrix},\ \boldsymbol{B}_1 = \begin{bmatrix} 0.8392 & 0.1338 \\ 0.6288 & 0.2071 \end{bmatrix},\ \boldsymbol{B}_2 = \begin{bmatrix} 1.0322 & 0.6298 \\ 1.0708 & 0.9778 \end{bmatrix}$$

with uncertainty modelled by (4.5), (6.27) and represented by

$$\boldsymbol{E}_{1b}=\begin{bmatrix} 0.1366 & 0.0186 \end{bmatrix}, \quad \boldsymbol{E}_{2b}=\begin{bmatrix} 0.0071 & 0.1225 \end{bmatrix},$$

$$\boldsymbol{E}_{1d}=\begin{bmatrix} 0.3043 & 0.0082 \\ 0.0079 & 0.0950 \end{bmatrix}, \quad \boldsymbol{E}_{2d}=\begin{bmatrix} 0.2935 & 0.1838 \\ 0.0288 & 0.3157 \end{bmatrix}$$

$$\boldsymbol{E}_1 = \begin{bmatrix} 0.0272 & 0.3127 \end{bmatrix}, \ \boldsymbol{E}_2 = \begin{bmatrix} 0.0129 & 0.3840 \end{bmatrix}, \ \boldsymbol{H} = \begin{bmatrix} 0.2257 \\ 0.0219 \end{bmatrix}$$

In this case, LMI (6.32) is feasible and the matrices are

$$\boldsymbol{W} = \begin{bmatrix} 2.7354 & -0.7082 \\ -0.7082 & 2.1651 \end{bmatrix}, \ \boldsymbol{Z} = \begin{bmatrix} 0.7730 & -0.1974 \\ -0.1974 & 0.7353 \end{bmatrix},$$

$$\boldsymbol{Z}_1=\begin{bmatrix} 0.7966 & -0.2018 \\ -0.2018 & 0.4507 \end{bmatrix}, \boldsymbol{Z}_2=\begin{bmatrix} 0.7883 & -0.1620 \\ -0.1620 & 0.5530 \end{bmatrix}, \boldsymbol{N}=\begin{bmatrix} 1.0029 & -0.5007 \\ -1.5858 & 1.1028 \end{bmatrix}$$

and scalar $\epsilon = 2.3197$ *which yields the stabilising matrix* $\boldsymbol{K}$ *equal to*

$$\boldsymbol{K} = \begin{bmatrix} 0.3351 & -0.1216 \\ -0.4893 & 0.3493 \end{bmatrix}$$

The resulting system is asymptotically stable independent of the delay sizes according to Theorem 6.4 with the following matrices

$$\boldsymbol{P}=\begin{bmatrix} 2.3227 & 0.6053 \\ 0.6053 & 3.0341 \end{bmatrix}, \quad \boldsymbol{Q}=\begin{bmatrix} 0.7478 & 0.3073 \\ 0.3073 & 0.9924 \end{bmatrix},$$

$$\boldsymbol{Q}_1=\begin{bmatrix} 0.6179 & 0.1094 \\ 0.1094 & 0.6936 \end{bmatrix}, \quad \boldsymbol{Q}_2=\begin{bmatrix} 0.6605 & 0.1537 \\ 0.1537 & 0.7948 \end{bmatrix}$$

and $\epsilon = 1.3709$ *computed.*

Remark 6.3. *In passing, the robust stability and stabilisation results given in this section can be extended naturally to the multiple delay case in a similar way as in Theorem 6.2.*

6.3. Concluding remarks

Time delays are usually result in unsatisfactory performance and are frequently a source of instability. Therefore their presence must be considered in realistic control design procedures for both 1-D and 2-D(n-D) systems. Due to the lack of results for 2-D systems with delays, this chapter provides the preliminary developments in this area. In particular, the implementable and computationally effective conditions for stability, robust stability and stabilisation of 2-D state-delayed systems are presented. Furthermore, it is shown that numerical procedures for asymptotic stability of n-D delay-free systems can be used for a stability investigation of 1-D multiple delay systems. However, the dimension of the matrices involved in the LMI based conditions could well be very large and therefore numerical difficulties could arise. This can occur, for example, when the system dimensionality is large ($n \gg 1$) and/or many delays are present. To overcome this potential problem, the approach based on EOA has been presented.

Chapter 7

CONCLUSIONS AND FUTURE WORKS

Several modern engineering fields such as image enhancement, signal and data processing or digital filtering, use 2-D(n-D) system theory due to the n-D character of considered processes and systems. Unfortunately, application of the classical (i.e. spectral) methods to analyse and synthesise n-D systems is a source of many computational problems, which make known computer-aided methods inefficient. Indeed, the number of system poles (variables) in the case of n-D systems can be infinite and no method exists to deal with such a big number of variables. Therefore the stability problem of n-D systems can be put into the class of undecidable problems and there is a need in the n-D system community for an appropriate theoretical framework which leads to easily implementable numerical methods to analyse and design n-D systems.

It turns out that among numerical techniques, convex and quasi-convex optimization methods which involve LMIs are the most promising, powerful tools for the analysis and design of control for 1-D systems. It is mainly because problems formulated in terms of LMI can be solved efficiently using a computer and they offer a framework for the formulation of problems arising in control.

In view of the described situation, the main purpose of this dissertation is to provide alternate problem formulations and the alternative, or sometimes only existing solutions to some theoretical problems for linear n-D and their class of great practical importance i.e. LRPs. The proposed approach is based on combining state-space representations of the considered class of n-D systems with the Lyapunov framework to derive the problem formulation in terms of BMI, which has been proven to be $\mathcal{NP}$-hard problems. However, some of the BMI problem formulations can be transformed into LMIs due to their "hidden" convexity properties. This is done using several techniques and further leads to exact or approximate solutions to the original problem.

Due to the fact that little or no work has yet been reported on differential LRPs, the great emphasis is put upon solving the problems of analysis and synthesis for this class of 2-D systems. The development of the presented method results in a computer numerical package which can assist with the analysis and design of considered classes of systems. Availability of such a software makes not only analysis and design for LRPs and n-D systems automated processes, but can overcome potentially difficult control problems for which traditional methods of analysis and synthesis may be limited. This is especially valid for systems where delays and/or parameter uncertainty appear.

The major developments presented in this dissertation can be summarized as follows:

- This dissertation provides computer implementable formulation of the following problems arising in analysis and synthesis 2-D systems and their classes i.e. LRPs.
  - The first results on robust stability and stabilisation of both differential and discrete LRPs allow us to consider the effect of uncertainty which occurred in system state-space models. It is shown that stability and stabilisation conditions which involve LMIs become easy to check with a computer. Further, it can be seen that it is possible to formulate optimization procedures which can be used to attenuate the effects of the uncertainty.
  - The numerical solution to the $\mathcal{H}_\infty$ control problems for LRPs are provided. The LMI formulation of such problems allows not only to provide numerical algorithms for controller design but also makes it possible to optimize some parameters of the design process. Moreover, the solution to the problem of the $\mathcal{H}_\infty$ output controller designing, where nominal computational complexity is very high, is given.
  - Derives the solution to the robust $\mathcal{H}_\infty$ control problems for LRPs. It is shown that LMI methods allow us to formulate and solve the $\mathcal{H}_\infty$ control problems for LRPs with parameter uncertainties.
  - Formulation and solving the $\mathcal{H}_2$ control problem for differential LRPs in spirit of LMI methods. It is proven that LMI techniques allow us to design the control laws which guarantee stability of the process and the maximum possible $\mathcal{H}_2$ disturbance attenuation.
  - Derives the solution to the guaranteed cost control problem. It is shown that control cost can be minimized with provided LMI conditions.
  - The last result shows that LMI methods can be applied to analyse and synthesise 2-D systems with delays. Moreover, it is proven that this result can be extended to deal with uncertain 2-D systems with delays.
- Application presented result to analyse and design two processes from computer engineering.
- Development of a Matlab-based tool which is a collection of Matlab functions to analyse and design considered systems and processes.

It should be pointed out that there are many research directions which should be pursued in order to improve and extend the result presented in this dissertation. In particular, the following research directions must be considered

- multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ control of LRPs and n-D systems,
- further development of LMI methods for systems with state and input delays,

- attempts to apply LMIs to the analysis of systems with state-dependent delays,
- application of LMI methods to filtering problems,
- generalizations for processes with nonlinearities and combinations with stochastic approaches.

It should be pointed out that it is possible to apply LMI methods to deal with a polynomial of two variables. As it is shown in (Henrion *et al.*, 2001) the stability test of a 2-D polynomial matrix in various regions of the complex plane can be cast as a non-convex rank-one LMI feasibility problem. Convex LMI relaxations can readily be derived from this formulation.

Another important issue to be addressed is the further development of the Matlab-based tool. This should be optimized to improve its efficiency, and extended to support a wider class of repetitive processes e.g. non-unit memory LRPs and (n-D) systems.

Streszczenie

Rozprawa dotyczy rozwiązywania problemów analizy i syntezy liniowych układów wielowymiarowych (*ang. multidimensional* – n-D), a w szczególności ich podklasy tj. liniowych procesów powtarzalnych (*ang. linear repetitive processes* – LRPs) z zastosowaniem współczesnych metod komputerowej analizy numerycznej.

Układy wielowymiarowe charakteryzują się występowaniem więcej niż jednej zmiennej niezależnej jako wyniku:

- występowania więcej niż jednej zmiennej przestrzennej,
- występowania wpływu przestrzeni i czasu,
- efektu czasowej/przestrzennej zmiennej oraz indeksu reprezentującego kolejną iterację, pas lub krok uczenia.

Generalnie układy wielowymiarowe, a w tym liniowe procesy powtarzalne znalazły zastosowanie w opisie wielu zjawisk i procesów występujących w licznych dziedzinach współczesnej techniki. Szczególnie interesującymi zastosowaniami są procesy iteracyjnego sterowania z uczeniem, iteracyjnego sterowania suboptymalnego bazującego na zasadzie maksimum, procesy przetwarzania równoległego i rozproszonego oraz sterowanie maszynami w górnictwie, hutnictwie, papiernictwie i rolnictwie. Wśród licznych zastosowań układów wielowymiarowych i ich klas znajdujemy również układy wielowymiarowego przetwarzania sygnałów i obrazów, kodowania i dekodowania oraz filtracji sygnałów, które często są używane w grafice komputerowej.

Należy jednak podkreślić, że zastosowanie modeli wielowymiarowych jest bardzo ograniczone, głównie ze względu na brak dobrze rozwiniętej teorii układów wielowymiarowych, która dostarczyłaby odpowiedniego formalizmu matematycznego, umożliwiającego zapisanie wielu problemów analizy i syntezy w postaci predestynującej do zastosowania szerokiego wachlarza efektywnych metod numerycznych. Efektywność jest tutaj rozumiana jako praktyczna możliwość rozwiązania rozważanego problemu w czasie wielomianowym tj. czas potrzebny do rozwiązania problemu jest ograniczony przez funkcję, która jest wielomianem zmiennej określającej wielkość zasobów potrzebnych do zdefiniowania problemu. Problem posiadający taką własność zaliczany jest do klasy problemów $\mathcal{P}$-trudnych.

W przypadku układów 1-D pokazano, iż wiele problemów analizy i syntezy może być sprowadzonych do postaci problemów badania własności pewnych wielomianów lub macierzy. Innymi słowy, problemy analizy i syntezy sprowadza się głównie do problemów związanych z wyznaczaniem i lokowaniem biegunów układu. Z kolei te problemy, mogą być rozwiązane przy użyciu algorytmów wielomianowych. Przykładem problemu, dla którego udowodniono istnienie algorytmu wielomianowego do jego rozwiązania jest problem stabilności. Istotnie, rozwiązanie

problemu stabilności może być sprowadzone do wyznaczenia wartości własnych macierzy systemowej i sprawdzenia czy wszystkie one mają moduły mniejsze od 1 (w przypadku układów dyskretnych) lub leżą w lewej półpłaszczyźnie zespolonej (dla układów ciągłych). Z kolei, wartości własne macierzy zawsze mogą być obliczone w czasie wielomianowym, dlatego problem stabilności należy do klasy problemów $\mathcal{P}$-trudnych. Należy tutaj zaznaczyć, że alternatywną metodą określania stabilności układu jest użycie kryterium Routh'a, które zawsze daje odpowiedź czasie wielomianowym. Oznacza to również, że istnienie wielomianowego algorytmu rozwiązującego problem analizy (stabilność) daje potencjalną możliwość rozwiązania problemu syntezy (stabilizacja). Co więcej, w przypadku układów 1-D, dla dużej liczby problemów zaliczanych do klasy problemów $\mathcal{NP}$-trudnych (czyli takich dla których nie udowodniono istnienia algorytmu wielomianowego) zostało przedstawionych wiele metod heurystycznych umożliwiających uzyskanie zadowalającego rozwiązania danego problemu.

Naturalnym jest postawienie pytania, czy problemy analizy i syntezy układów wielowymiarowych mogą być równie efektywnie rozwiązane jak to ma miejsce w przypadku układów jednowymiarowych. Niestety, okazuje się, że zastosowanie znanych (tj. stosowanych w teorii układów 1-D) i efektywnych metod do rozwiązywania problemów analizy i syntezy układów wielowymiarowych jest bardzo ograniczone, a często nawet niemożliwe. Trudności te są przede wszystkim związane z brakiem lub wysokim stopniem komplikacji istniejącego formalizmu matematycznego, uniemożliwiającym zastosowanie efektywnych metod w celu rozwiązania problemu zapisanego z użyciem tego właśnie formalizmu.

Najbardziej istotnym faktem, związanym z zastosowaniem metod opartych na manipulowaniu biegunami układu wielowymiarowego, jest możliwość istnienia nieskończenie wielu biegunów układu. Dlatego też, w kontekście złożoności obliczeniowej, problem analizy tychże układów zaliczany jest do klasy problemów $\mathcal{NP}$-trudnych lub nawet nierozstrzygalnych, gdyż trudno jest testować położenie każdego bieguna układu (gdy ich liczba dąży do nieskończoności), co jest odpowiednikiem tradycyjnego spektralnego warunku stabilności układu. Dodatkowo należy pamiętać, że w przypadku układów 1-D położenie biegunów całkowicie określa dynamikę układu i wiemy w jakie miejsca płaszczyzny zespolonej przesuwać bieguny, aby zapewnić sobie określoną dynamikę. Dla układów wielowymiarowych nie dysponujemy taką wiedzą, możemy tylko, podobnie jak dla przypadku 1-D, scharakteryzować stabilność i ewentualnie marginesy stabilności w terminach biegunów.

Ważnymi kwestiami, dominującymi ostatnio w teorii sterowania, są odporność układów na niepewności i zakłócenia oraz sterowanie optymalne przy zadanych wskaźnikach jakości. Dlatego konieczne stało się również rozważanie tych kwestii dla układów wielowymiarowych, a w szczególności dla liniowych procesów powtarzalnych, dla których odczuwalny jest brak wyników w tym zakresie. Jednakże, okazało się, że badanie odporności układów i procesów w przypadkach występowania niepewności parametrów oraz zakłóceń jest zadaniem trudnym z punktu widzenia informatyki. Trudności te wynikają z bardzo dużej złożoności obliczeniowej tychże problemów (zakłada się, że są to problemy $\mathcal{NP}$-trudne). Dlatego wciąż

próbuje się sprowadzić większość problemów analizy i syntezy do postaci umożliwiającej zastosowanie efektywnych algorytmów, które pozwoliłyby na podanie jakichkolwiek rozwiązań (gdy występuje ich brak) lub umożliwienie zredukowania konserwatyzmu istniejących już wyników.

Ze względu na dużą liczbę potencjalnych zastosowań liniowych procesów powtarzalnych oraz układów n-D, możliwość sprowadzenia problemów analizy i syntezy tychże układów, w szczególności dla przypadków kiedy występują niepewności, zakłócenia i opóźnienia, do postaci pozwalającej na zastosowanie efektywnych i znanych procedur numerycznych, stała się istotnym zagadnieniem.

Głównym celem pracy jest zatem zastosowanie liniowych nierówności macierzowych (*ang. linear matrix inequalities* – LMI) do sformułowania i rozwiązania wielu problemów z zakresu sterowania układów wielowymiarowych i ich klas tj. liniowych procesów powtarzalnych celem ich komputerowej analizy i syntezy. Atrakcyjność proponowanego podejścia związana jest z istnieniem tzw. algorytmów punktu wewnętrznego, posiadających złożoność wielomianową, które służą do rozwiązania problemów optymalizacyjnych z ograniczeniami w postaci liniowych nierówności macierzowych. Dodatkowym atutem takiego postępowania jest możliwość przezwyciężenia wielu problemów występujących przy zastosowaniu klasycznych metod analizy i syntezy dla modeli zawierających niepewności oraz opóźnienia, gdyż unikamy bezpośredniego wyznaczania i manipulowania biegunami układu wielowymiarowego. Choć otrzymane warunki istnienia rozwiązania są tylko warunkami wystarczającymi, to jednak najczęściej stanowią one jedyne znane rozwiązanie. Kwestia ta ma szczególne znaczenie dla liniowych różniczkowych procesów powtarzalnych, dla których tylko niewielka liczba rezultatów została opublikowana.

W celu sformułowania rozważanych problemów analizy i syntezy układów wielowymiarowych w formie liniowych nierówności macierzowych, w pracy zastosowano następujące podejście. Po pierwsze, użyto metod Lapunowa uzyskując problem w postaci nierówności macierzowej, gdzie występują znane i poszukiwane macierze. Nierówności te, najczęściej są nieliniowe względem poszukiwanych parametrów, co implikuje brak efektywnych algorytmów do ich rozwiązania. Należy jednak podkreślić, że w wielu przypadkach możliwa jest eksploracja pewnych własności otrzymanych nierówności w celu otrzymania postaci liniowej nierówności macierzowej. Szczególną uwagę poświęcono problemom analizy i syntezy układów z niepewnościami parametrów, których większość zaliczana jest do klasy problemów $\mathcal{NP}$-trudnych. Jednakże, możliwe jest uzyskanie przybliżonego rozwiązania takiego problemu przy zastosowaniu metod liniowych nierówności macierzowych.

Wszystkie warunki istnienia rozwiązania rozważanych w pracy problemów zostały zaprezentowanie w formie liniowych nierówności macierzowych, umożliwiając implementację i rozwiązanie z użyciem dostępnego oprogramowania. Przykłady procedur, stanowiących część opracowanego pakietu numerycznego zostały również zawarte w pracy. Dodatkowo, opracowane i zaimplementowane procedury numeryczne zostały zastosowane do analizy praktycznych problemów znajdujących się w obszarze zainteresowań informatyki.

BIBLIOGRAPHY

Ackerman J. (1997): *Robust Control. Systems with Uncertain Physical Parameters.* — Communications and Control Engineering Series, 3rd edn, London, England: Springer-Verlag.

Agathoklis P. (1988): *Lower bounds for the stability margin of discrete two-dimensional systems based on the two-dimensional Lyapunov equation.* — IEEE Transactions on Circuits and Systems, Vol. 35, No. 6, pp. 745–749.

Agathoklis P. and Foda S. (1989): *Stability and the matrix Lyapunov equation for delay differential systems.* — International Journal of Control, Vol. 49, No. 2, pp. 417–432.

Amann N. (1996): *Optimal algorithms for iterative learning control.* — PhD thesis, University of Exeter, England.

Amann N., Owens D. H. and Rogers E. (1998): *Predictive optimal iterative learnig control.* — International Journal of Control, Vol. 69, No. 2, pp. 203–226.

Andersen E. (2001): *MOSEK Manual.* — MOSEK ApS, Copenhagen, Denmark. Available at http://www.mosek.com/

Apkarian P. and Tuan H. D. (1998): *Parameterized LMIs in control theory.* — Proc. 37th IEEE Conf. Decision and Control (CDC), Tampa, USA, 16-18 December, 1998, Vol. 1, pp. 152–157.

Apkarian P. and Tuan H. D. (2000): *Parameterized LMIs in control theory.* — SIAM Journal of Control and Optimization, Vol. 38, No. 4, pp. 1241–1264.

Attasi S. (1973): *Systèmes lineaires homogènes à deux indices.* — Rapport Labria 31, IRIA.

Bakshee I. (2003): *Mathematica. Control System Professional.* — Wolfram Research Inc., Champaign, USA.

Basu S. (2002): *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.

Bauer P. H., Sichitiu M. and Premaratne K. (2001): *Stability of 2-D processes with time-variant communication delays.* — Proc. IEEE Int. Symp. Circuits and Systems (ISCAS), Sydney, Australia, 6-9 May, 2001, Vol. 2, pp. 497–500.

Beck C. (1991): *Computational issues in solving LMIs.* — Proc. 30th Conf. Decision and Control (CDC), Brighton, England, 11-13 December, 1991, Vol. 2, pp. 1259–1260.

Benson S. (2003): *Parallel computing on semidefinite programs.* — Research report ANL/MCS-P939-0302, Mathematics and Computer Science Division, Argonne National Laboratory. Available at http://www-unix.mcs.anl.gov/~benson/dsdp/

Benson S. and Ye Y. (2002): *DSDP4 - a software package implementing the dual-scaling algorithm for semidefinite programming.* — Research report ANL/MCS-TM-255, Mathematics and Computer Science Division, Argonne National Laboratory.

Benton S. E. (2000): *Analysis and control of linear repetitive processes.* — PhD thesis, University of Southampton, Southampton, England.

Bliman P.-A. (2002): *Lyapunov equation for the stability of 2-D Systems.* — Multidimensional Systems and Signal Processing, Vol. 13, No. 2, pp. 201–222.

Blondel V. and Tsitsiklis J. N. (1997): *NP-hardness of some linear control design problems.* — SIAM Journal on Control and Optimization, Vol. 35, No. 6, pp. 2118–2127.

Blondel V. and Tsitsiklis J. N. (2000a): *The boundedness of all products of a pair of matrices is undecidable.* — Systems and Control Letters, Vol. 41, No. 2, pp. 135–140.

Blondel V. and Tsitsiklis J. N. (2000b): *A survey of computational complexity results in systems and control.* — Automatica, Vol. 36, No. 9, pp. 1249–1274.

Bose N. K. (1982): *Applied Multidimensional Systems Theory.* — New York, USA: Van Nostrand-Reinhold.

Bose N. K. (1985): *Multidimensional Systems Theory. Progres, Directions and Open Probles in Multidimensional Systems.* — Dordrecht, Holland: D. Reidel Publishing Company.

Bose N. K. (2001): *Two decades of multidimensional research and future trends.* — in K. Gałkowski and J. Wood (eds), Mutidimensional Signals, Circuits and Systems, Systems and Control Book Series, London, England: Taylor and Francis, pp. 5–27.

Bose N. K. (ed.) (1977): *Multidimensional Systems.* — Proceedings of the IEEE, Vol. 65, No. 6.

Boukas E. K. and Liu Z. K. (2003): *Deterministic and Stochastic Time Delay Systems.* — Control Engineering, Boston, USA: Birkhauser.

Boyd S., Ghaoui L. E., Feron E. and Balakrishnan V. (1994): *Linear Matrix Inequalities in System and Control Theory.* — SIAM Studies in Applied and Numerical Mathematics, Philadelphia, USA: SIAM, Vol. 15.

Boyd S. and Vandenberghe L. (2004): *Convex Optimization.* — Cambridge, England: Cambridge University Press.

Bracewell R. N. (1995): *Two-dimensional Imaging.* — Prentice Hall Signal Processing Series, Upper Saddle River, USA: Prentice Hall Inc.

Buchberger B. (1985): *Gröbner bases: an algorithmic method in polynomial ideal theory.* — in N. K. Bose (ed.), Multidimensional Systems Theory, Dordrecht, Holland: Reidel Publishing Company, pp. 184–232.

Chen Y. and Wen C. (1999): *Iterative Learning Control. Convergence, Robustness and Applications.* — Lecture Notes in Control and Information Sciences, London, England: Springer-Verlag, Vol. 248.

Chiasson J. N., Brierley S. D. and Lee E. B. (1985): *A simplified derivation of the Zeheb-Walch 2-D stability test with applications to time-delay systems.* — IEEE Transactions on Automatic Control, Vol. 30, No. 4, pp. 411–414.

Cook P. A. (2000): *Stability of two-dimensional feedback systems.* — International Journal of Control, Vol. 73, No. 4, pp. 343–348.

Curtin E. and Saba S. (1999): *Stability and margin of stability tests for multidimensional filters.* — IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 46, No. 7, pp. 806–809.

D'Andrea R. (1999): *The Multidimensional System Toolbox. Version 1.0.* — Research report, Mechanical and Areospace Engineering. Cornell University. Available at http://www.mae.cornell.edu/raff/Software/MDtoolbox/mdtoolbox.pdf

Doyle J. C., Glover K., Khargonekar P. P. and Francis B. A. (1989): *State-space solutions to standard H_2 and H_∞ control problems.* — IEEE Transactions on Automatic Control, Vol. 34, No. 8, pp. 831–847.

Du C. and Xie L. (1999a): *LMI approach to output feedback stabilization of 2-D discrete systems.* — International Journal of Control, Vol. 72, No. 2, pp. 97–106.

Du C. and Xie L. (1999b): *Stability analysis and stabilization of uncertain two-dimensional discrete systems: an LMI approach.* — IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 46, No. 11, pp. 1371–1374.

Du C. and Xie L. (2002): *H_∞ Control and Filtering of Two-dimensional Systems.* — Lecture Notes in Control and Information Sciences, Berlin, Germany: Springer-Verlag, Vol. 278.

Du C., Xie L. and Soh Y. C. (2000): *H_∞ filtering of 2-D discrete systems.* — IEEE Transactions on Signal Processing, Vol. 48, No. 6, pp. 1760–1768.

Du C., Xie L. and Zhang C. (2001): *H_∞ control and robust stabilization of two-dimensional systems in Roesser models.* — Automatica, Vol. 37, No. 2, pp. 205–211.

Dudgeon D. E. and Merserau R. M. (1984): *Multidimensional Digital Signal Processing.* — Prentice-Hall Signal Processing Series, Englewood Cliffs, USA: Prentice Hall.

Dugard L. and Verriest E. I. (1998): *Stability and Control of Time-delay Systems.* — Lecture Notes in Control and Information Sciences, London, England: Springer-Verlag, Vol. 228.

Dullerud G. E. and Paganini F. (2000): *A Course in Robust Control Theory. A Convex Approach.* — Texts in Applied Mathematics, New York, USA: Springer-Verlag, Vol. 36.

El Ghaoui L. (eds) and Niculescu S.-I. (1999): *Advances in Linear Matrix Inequality Methods in Control.* — Advances in Design and Control, Philadelphia, USA: SIAM, Vol. 2.

Fernando T. and Trinh H. (1999): *Lower bounds for stability margin of two-dimensional discrete systems using the MacLaurine series.* — Computers and Electrical Engineering, Vol. 25, No. 2, pp. 95–109.

Foda S. and Agathoklis P. (1992): *Control of the metal rolling process: a multidimensional system approach.* — Journal of the Franklin Institute, Vol. 329, No. 2, pp. 317–332.

Fornasini E. and Marchesini G. (1976): *State-space realization theory of two-dimensional filters.* — IEEE Transactions on Automatic Control, Vol. 21, No. 4, pp. 484–492.

Fornasini E. and Marchesini G. (1978): *Doubly indexed dynamical systems: state models and structural properties.* — Mathematical Systems Theory, Vol. 12, pp. 59–72.

Fu M. and Luo Z.-Q. (1997): *Computational complexity of a problem arising in fixed order output feedback design.* — Systems and Control Letters, Vol. 30, No. 5, pp. 209–215.

Fujisawa K., Kojima M. and Nakata K. (2000): *SDPA (Semidefinite Programming Algorithm) - user's manual.* — Technical Report B-308, Tokyo Institute of Technology. Available at http://rnc.r.dendai.ac.jp/~fujisawa/sdpa_doc.pdf

Gahinet P. and Apkarian P. (1994): *A linear matrix inequality approach to H_∞ control.* — International Journal of Robust and Nonlinear Control, Vol. 4, pp. 421–448.

Gahinet P. and Nemirovski A. (1997): *The projective method for solving linear matrix inequalities.* — Mathematical Programming, Vol. 77, No. 2, pp. 163–190.

Gahinet P., Nemirovski A., Laub A. J. and Chilali M. (1995): *LMI Control Toolbox for use with MATLAB.* — The Mathworks Partner Series, Natick, USA: The MathWorks Inc.

Gałkowski K. (2001a): *State-Space Realisations of Linear 2-D Systems with Extensions to the General nD Case.* — Lecture Notes in Control and Information Sciences, London, England: Springer-Verlag, Vol. 263.

Gałkowski K., Lam J., Rogers E., Xu S., Sulikowski B., Paszke W. and Owens D. H. (2003a): *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.

Gałkowski K., Lam J., Xu S. and Lin Z. (2003b): *LMI approach to state-feedback stabilization of multidimensional systems.* — International Journal of Control, Vol. 76, No. 14, pp. 1428–1436.

Gałkowski K., Paszke W., Lam J., Rogers E. and Owens D. H. (2002a): *Stability of uncertain differential linear repetitive processes.* — Proc. XIV Nat. Conf. Automatic Control (KKA), Zielona Góra, Poland, 24-27 June, 2002, Vol. 1, pp. 297–302.

Gałkowski K., Paszke W., Rogers E., Xu S., Lam J. and Owens D. H. (2003c): *Stability and control of differential linear repetitive processes using an LMI seting.* — IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing, Vol. 50, No. 9, pp. 662–666.

Gałkowski K., Paszke W., Sulikowski B., Rogers E. and Owens D. H. (2002b): *LMI based stability analysis and controller design for a class of 2D continuous-discrete linear systems.* — Proc. American Control Conference (ACC), Anchorage, USA, 8-10 May, 2002, Vol. 1, pp. 29–34.

Gałkowski K., Rogers E., Gramacki A., Gramacki J. and Owens D. H. (2000): *Development of a Matlab Toolbox for a class of 2D linear systems.* — SAMS /Systems Analysis - Modelling - Simulation/ special issue on Analysis and Control of Technological Systems, Vol. 38, pp. 313–324.

Gałkowski K., Rogers E., Gramacki A., Gramacki J. and Owens D. H. (2001b): *Stability and dynamic boundary condition decoupling analysis for a class of 2-D discrete linear systems.* — IEE Proceedings - Circuits Devices Systems, Vol. 148, No. 3, pp. 126–134.

Gałkowski K., Rogers E., Paszke W. and Owens D. H. (2003d): *Linear repetitive process control theory applied to a physical example.* — International Journal of Applied Mathematics and Computer Science, Vol. 13, No. 1, pp. 87–99.

Gałkowski K., Rogers E., Wood J., Benton S. E. and Owens D. H. (2002c): *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.

Gałkowski K., Rogers E., Xu S., Lam J. and Owens D. H. (2002d): *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.

Gałkowski K., Sulikowski B. and Paszke W. (2003e): *Multidimesional systems.* — Pomiary Automatyka Kontrola, No. 2-3, pp. 53–58. (in Polish)

Gałkowski K. and Wood J. (eds) (2001): *Multidimensional Signals, Circuits and Systems.* — Systems and Control Book Series, London, England: Taylor and Francis.

Geng Z., Carroll R. and Xie J. (1990): *Two-dimensional model and algorithm analysis for a class of iterative learning control.* — International Journal of Control, Vol. 52, No. 4, pp. 833–862.

Goh K. C., Turan L., Safonov M. G., Papavassilopoulos G. P. and Ly J. H. (1994): *Biaffine matrix inequality properties and computional methods.* — Proc. American Control Conference (ACC), Baltimore, USA, 29 June - 1 July, 1994, pp. 850–855.

Golub G. M. and Loan C. V. (1996): *Matrix Computations.* — John Hopkins Series in the Mathematical Sciences, 3rd edn, Baltimore, USA: The John Hopkins University Press.

Gomez C. (1999): *Engineering and Scientific Computing with Scilab.* — Boston, USA: Birkhäuser.

Górecki H., Fuksa S., Grabowski P. and Koryłowski A. (1989): *Analysis and Synthesis of Time-delay Systems.* — New York, USA: John Wiley and Sons.

Gramacki A. (1999a): *Discretisation methods of linear differential repetitive processes.* — PhD thesis, Faculty of Electrical Engineering, Technical University of Zielona Góra, Zielona Góra, Poland. (in Polish)

Gramacki J. (1999b): *Stability and stabilisation linear repetitive processes.* — PhD thesis, Faculty of Electrical Engineering, Technical University of Zielona Góra, Zielona Góra, Poland. (in Polish)

Guan X., Long C. and Duan G. (2001): *Robust optimal guaranteed cost control for 2D discrete systems.* — IEE Proceedings - Control Theory Applications, Vol. 148, No. 5, pp. 355–361.

Hale J. K. and Lunel S. M. V. (1993): *Introduction to Functional Differential Equations.* — Applied Mathematical Sciences, New York, USA: Springer-Verlag, Vol. 99.

Handkiwicz A., Kropidłowski M. and Łukowiak M. (2000): *Two-dimensional signal processing based on FPGA.* — Proc. 2nd Int. Workshop on Multidimensional (nD) Systems, Czocha Castle, Lower Silesia, Poland, 27 - 30 June, 2000, pp. 143–148.

Hätönen J. and Ylinen R. (2003): *Polynomial systems theory applied to the analysis and design of multidimensional systems.* — International Journal of Applied Mathematics and Computer Science, Vol. 13, No. 1, pp. 15–29.

Helton J. W. and Merino O. (1998): *Classical Control Using H_∞ Methods. Theory, Optimization, and Design.* — Philadelphia, USA: SIAM.

Henrion D. (2003): *Course on LMI Optimization with applications in control.* — Prague, Czech Republic. Available at http://www.laas.fr/~henrion/courses/lmi

Henrion D., Ŝebek M. and Bachelier O. (2001): *Rank-one LMI approach to stability of 2-d polynomial matrices.* — Multidimensional Systems and Signal Processing, Vol. 12, No. 1, pp. 33–48.

Higham N. J., Konstantinov M., Mehrmann V. and Petkov P. (2004): *The sensitivity of computational control problems.* — IEEE Control Systems Magazine, Vol. 24, No. 1, pp. 28–43.

Hinamoto T. (1989): *The Fornasini-Marchesini model with no overflow oscillations and its application to 2-D digital filter design.* — Proc. IEEE Int. Symp. Circuits and Systems (ISCAS), Portland, USA, 9-11 May, 1989, Vol. 3, pp. 1680–1683.

Hinamoto T. (1997): *Stability of 2-D discrete systems described by the Fornasini-Marchesini second model.* — IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 44, No. 3, pp. 254–257.

Iwasaki T. and Skelton R. E. (1994): *All controlers for the general H_∞ control problem: LMI existence conditions and space state formulas.* — Automatica, Vol. 30, No. 8, pp. 1307–1317.

Iwasaki T. and Skelton R. E. (1995a): *The XY-centering algorithm for the dual LMI problem: a new approach to fixed order control design.* — International Journal of Control, Vol. 62, No. 6, pp. 1257–1272.

Iwasaki T. and Skelton R. E. (1995b): *A unified approach to fixed order controller design via linear matrix inequalities.* — Mathematical Problems in Engineering, Vol. 1, No. 1, pp. 59–75.

Jury E. (1978): *Stability of multidimensional scalar and matrix polynomial.* — Proceedings of the IEEE, Vol. 66, No. 9, pp. 1018–1047.

Kaczorek T. (1985): *Two-dimensional Linear Systems.* — Lecture Notes in Control and Information Sciences, Berlin, Germany: Springer-Verlag, Vol. 68.

Kaczorek T. (1994): *Singular 2D continuous-discrete linear systems.* — Bulletin Polish Academy of Science. Technical Sciences, Electronics and Electrotechnics, Vol. 42, pp. 417–422.

Kalker T., Park H. and Vetterli M. (1995): *Gr oebner basis techniques in multidimensional multirate systems.* — Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), Detroit, USA, 9-12 May, 1995, Vol. 4, pp. 2121–2124.

Kamen E. W. (1980): *On the relationship between zero criteria for two-variable polynomials and asymptotic stability of delay differential equations.* — IEEE Transactions on Automatic Control, Vol. 25, No. 5, pp. 983–984.

Khachiyan L. (1979): *A polynomial algorithm in linear programming.* — Soviet Mathematics Doklady, Vol. 20, pp. 191–194.

Khargonekar P. P., Petersen I. R. and Zhou K. (1990): *Robust stabilization of uncertain linear systems: quadratic stabilizability and H_∞ control theory.* — IEEE Transactions on Automatic Control, Vol. 35, No. 3, pp. 356–361.

Kharitonov V. L. and Zhabko A. P. (2003): *Lyapunov-Krasovskii approach to the robust stability analysis of time-delay systems.* — Automatica, Vol. 39, No. 1, pp. 15–20.

Kolmanovskii V. and Myshkis A. (1999): *Itroduction to the Theory and Applications of Functional Differential Equations.* — Mathematics and Its Applications, Dordrecht, The Netherlands: Kluver Academic Publishers, Vol. 463.

Ko cvara M. and Stingl M. (2003): *PENNON - a code for convex nonlinear and semidefinite programming.* — Optimization Methods and Software, Vol. 18, No. 3, pp. 317–330.

Kung S.-Y., L evy B., Morf M. and Kaliath T. (1977): *New results in 2-D systems theory, part II: 2-D state-space models - realization and the notions of controllability, observability and minimality.* — Proceedings of the IEEE, Vol. 65, No. 6, pp. 945–961.

Kurek J. E. (1985): *The general state-space model for a two-dimensional linear digital system.* — IEEE Transactions on Automatic Control, Vol. 30, No. 6, pp. 600–602.

Kurek J. E. and Zaremba M. B. (1993): *Iterative learnig control synthesis based on 2-D system theory.* — IEEE Transactions on Automatic Control, Vol. 38, No. 1, pp. 121–125.

Lee Y. S. and Kwon W. H. (2002): *Delay-dependent robust stabilization uncertain discrete-time state-delayed systems.* — Proc. 15th Triennial World Congress of the IFAC, Barcelona, Spain, 2002. (CD-ROM)

Lin Z. (2001): *Output feedback stabilizability and stabilization of linear n-D systems.* — in K. Gałkowski and J. Wood (eds), Mutidimensional Signals, Circuits and Systems, Systems and Control Book Series, London, England: Taylor and Francis, pp. 59–76.

Lin Z., Lam J., Gałkowski K. and Xu S. (2001): *A constructive approach to stabilizability and stabilization of a class of nD systems.* — Multidimensional Systems and Signal Processing, Vol. 12, No. 3-4, pp. 329–343.

Löfberg J. (2004): *YALMIP 3.* — Available at http://control.ee.ethz.ch/~joloef/yalmip.msql

Loiseau J. J. and Brethé D. (1997): *The use of 2-D system theory for the control of time-delay systems.* — Journal Europeen des Systemes Automatises, Vol. 31, pp. 1043–1058.

Longman R. W. (2000): *Iterative learning control - dynamic systems that learn in time and repetitions.* — Proc. 2nd Int. Workshop on Multidimensional (nD) Systems, Czocha Castle, Lower Silesia, Poland, 27-30 June, 2000, pp. 55–65.

Lu W.-S. (1994): *On a Lyapunov approach to stability analysis of 2-D digital filters.* — IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 41, No. 10, pp. 665–669.

Lu W.-S. (2002): *A unified approach for the design of 2-D digital filters via semidefinite programming.* — IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 49, No. 6, pp. 814–826.

Lu W.-S. and Antoniou A. (1992): *Two-dimensional Digital Filters.* — Electrical Engineering and Elecronics, New York, USA: Marcel Dekker,Vol. 80.

Lu W.-S. and Lee E. (1985): *Stability analysis for two-dimensional systems via a Lyapunov approach.* — IEEE Transactions on Circuits and Systems, Vol. 32, No. 1, pp. 61–68.

Lustig L. and Rothberg E. (1996): *Gigaflops in linear programming.* — Operations Research Letters, Vol. 18, No. 4, pp. 157–165.

Mahmoud M. S. (2000): *Robust Control and Filtering for Time-delay Systems.* — Control Engineering Series, New York, USA: Marcel Dekker, Vol. 5.

Malakorn T. (2003): *Multidimensional linear systems and robust control.* — PhD thesis, Faculty of the Virginia Polytechnic Institute, Blacksburg, USA.

Malek-Zavarei M. and Jamshidi M. (1987): *Time-delay Systems: Analysis, Optimization and Applications.* — North Holland Systems and Control, New York, USA: North-Holland, Vol. 9.

Meinsma G. (1997): *Interior Point Methods - Mini Course.* — The University of Twente, Enschede, The Netherlands. Available at http://wwwhome.math.utwente.nl/~meinsmag/courses/ipm.html

Mese M. and Vaidyanathan P. P. (2002): *Recent advances in digital halftoning and inverse halftoning methods.* — IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 49, No. 6, pp. 790–805.

Miri S. A. and Aplevich J. D. (2000): *Iterative techniques for decoding n-dimensional codes.* — Proc. 2nd Int. Workshop on Multidimensional (nD) Systems, Czocha Castle, Lower Silesia, Poland, 27-30 June, 2000, pp. 155–158.

Mittelmann H. (2000): *An independent benchmarking of SDP and SOCP solvers.* — Mathematical Programming, Vol. 95, No. 2, pp. 407–430.

Moheimani S. O. R. and Petersen I. R. (1996): *Optimal guaranteed cost control of uncertain systems via static and dynamic output feedback.* — Automatica, Vol. 32, No. 4, pp. 575–579.

Moon Y. S., Park P., Kwon W.-H. and Lee Y.-S. (2001): *Delay-dependent robust stabilisation of uncertain state-delayed systems.* — International Journal of Control, Vol. 74, No. 14, pp. 1447–1455.

Moore K. L. (1983): *Iterative learning control for deterministic systems.* — Advances in Indrustrial Control Series, London, England: Springer-Verlag.

Nemirovskii A. and Gahinet P. (1994): *The projective method for solving linear matrix inequalities.* — Proc. American Control Conference (ACC), Balimore, USA, 29 June - 1 July, 1994, Vol. 1, pp. 840–844.

Nesterov Y. and Nemirovskii A. (1994): *Interior-point Polynomial Algorithms in Convex Programming.* — SIAM Studies in Applied and Numerical Mathematics, Philadelphia, USA: SIAM, Vol. 13.

Niculescu S.-I. (2001): *Delay Effects on Stability.* — Lecture Notes in Control and Information Sciences, London, England: Springer-Verlag, Vol. 269.

Oliveira M. C. d. (2002): *Linear systems control and LMIs. Lecture Notes of University of Campinas.* — Campinas, Brasil. Available at http://www.dt.fee.unicamp.br/∼mauricio/

Oliveria M. C. d. and Geromel J. C. (1997): *Numerical comparison of output feedback design methods.* — Proc. American Control Conference (ACC), Albuquerque, USA, 4-6 June, 1997, Vol. 1, pp. 72–76.

Owens D. H., Amann N., Rogers E. and French M. (2000): *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.

Packard A., Zhou K., Pandey P. and Becker G. (1991): *A collection of robust control problems leading to LMI's.* — Proc. 30th Conf. Decision and Control (CDC), Brighton, England, 11-13 December, 1991, Vol. 2, pp. 1245–1250.

Paszke W., Gałkowski K., Rogers E. and Owens D. H. (2003): *H_∞ control of discrete linear repetitive processes.* — Proc. 42th IEEE Conf. Decision and Control (CDC), Hyatt Regency Maui, USA, 9-12 December, 2003, Vol. 1, pp. 628–633.

Paszke W., Gałkowski K., Rogers E. and Owens D. H. (2004): *H_∞ control of differential linear repetitive processes.* — Proc. American Control Conference (ACC), Boston, USA, 30 June - 2 July, 2004, Vol. 2, pp. 1386–1391.

Paszke W., Lam J., Gałkowski K. and Rogers E. (2003): *H_∞ control of 2D linear state-delayed systems.* — Proc. 4th IFAC Workshop on Time Delay Systems (TDS), Paris, France, 8-10 September, 2003. (CD-ROM)

Paszke W., Lam J., Gałkowski K., Xu S. and Lin Z. (2004): *Robust stability and stabilisation of 2D discrete state-delayed systems.* — System and Control Letters, Vol. 51, No. 3-4, pp. 277–291.

Petersen I. R. and McFarlane D. C. (1994): *Optimal guaranteed cost control and filtering for uncertain linear systems.* — IEEE Transactions on Automatic Control, Vol. 39, No. 9, pp. 1971–1977.

Petersen I. R., McFarlane D. C. and Rotea M. A. (1998): *Optimal guaranteed cost control of discrete-time uncertain linear systems.* — International Journal of Robust and Nonlinear Control, Vol. 8, No. 8, pp. 649–657.

Pillai H., Wood J. and Rogers E. (2002): *On homomorphisms of n-D behaviors.* — IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 49, No. 6, pp. 732–742.

Polderman J. W. and Willems J. C. (1997): *Introduction to Mathematical Systems Theory. A Behavioral Approach.* — Texts in Applied Mathematics, New York, USA: Springer-Verlag, Vol. 99.

Richard J.-P. (2003): *Time-delay systems: an overview of some recent advances and open problems.* — Automatica, Vol. 39, No. 10, pp. 1667–1694.

Roberts P. D. (2000a): *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.

Roberts P. D. (2000b): *Stability analysis of iterative optimal control algorithms modelled as linear unit memory repetitive processes.* — IEE Proceedings - Control Theory Applications, Vol. 147, No. 3, pp. 229–237.

Roberts P. D. (2002): *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.

Roesser R. P. (1975): *A discrete state-space model for linear image processing.* — IEEE Transactions on Automatic Control, Vol. 20, No. 1, pp. 1–10.

Rogers E., Gałkowski K., Gramacki A., Gramacki J. and Owens D. H. (2002): *Stability and controllability of class of 2-D linear systems with dynamic boundary conditions.* — IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 49, No. 2, pp. 181–195.

Rogers E., Gałkowski K. and Owens D. H. (2005): *Control Systems Theory and Applications for Linear Repetitive Processes.* — Lecture Notes in Control and Information Sciences, Springer-Verlag. (to be published)

Rogers E. and Owens D. H. (1992): *Stability Analysis for Linear Repetitive Processes.* — Lecture Notes in Control and Information Sciences, London, England: Springer-Verlag, Vol. 175.

Rogers E. and Owens D. H. (2001): *Two decades of research on linear repetitive processes.* — in K. Gałkowski and J. Wood (eds), Mutidimensional Signals, Circuits and Systems, Systems and Control Book Series, London, England: Taylor and Francis, pp. 107–147.

Saberi A., Sanuti P. and Chen B. M. (1995): *H_2 Optimal Control.* — Prentice-Hall International Series in System and Control Engineering, London, England: Prentice-Hall International.

Safonov M., Goh K. C. and Ly J. H. (1994): *Control system synthesis via bilinear matrix inequalities.* — Proc. American Control Conference (ACC), Baltimore, USA, 29 June - 1 July, 1994, Vol. 1, pp. 45–49.

Scherer C. and Weiland S. (2002): *DISC course linear matrix inequalities in control.* — Delft University of Technology, Delft, The Netherlands. Available at http://www.cs.ele.tue.nl/sweiland/lmi.htm

Ŝebek M. (1993): *Sub-optimum H_∞ problem for 2-D MIMO systems.* — Proc. 2nd European Control Conference (ECC), Groningen, The Netherlands, 28 June - 1 July, 1993, pp. 1476–1479.

Shankar S. and Willems J. (2000): *Behaviors of n-D systems.* — Proc. 2nd Int. Workshop on Multidimensional (nD) Systems, Czocha Castle, Lower Silesia, Poland, 27-30 June, 2000, pp. 23–30.

Shi Y. Q. and Zhang X. M. (2002): *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.

Skelton R. E., Iwasaki T. and Grigoriadis K. (1998): *A Unified Algebraic Approach to Linear Control Design.* — Series in Systems and Control, London, England: Taylor and Francis.

Srikant R. (2004): *The Mathematics of Internet Congestion Control.* — Systems & Control: Foundations & Applications, Boston, USA: Brickhauser.

Stoorvogel A. (1992): *The H_∞ Control Problem. A State Space Approach.* — Prentice-Hall International Series in System and Control Engineering, London, England: Prentice-Hall International.

Sturm J. F. (1999): *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones.* — Optimization Methods and Software, Vol. 11, No. 1, pp. 625–653. Special issue on Interior Point Methods (CD supplement with software)

Sturm J. F. (2000): *Central Region Method.* — in J. Frenk, C. Roos, T. Terlaky and S. Zhang (eds), High performance optimization, kluwer academic publishers, pp. 157–194.

The Mathworks Inc. (2002): *Control System Toolbox user's guide.* — The Mathworks Inc., Natick, USA. Available at http://www.mathworks.com/access/helpdesk/help/helpdesk.html

The Mathworks Inc. (2004): *Matlab. The Language of Technical Computing.* — The Mathworks Inc., Natick, USA. Available at http://www.mathworks.com/access/helpdesk/help/helpdesk.html

Tian C. and Zhang J. (2004): *Exponential asymptotic stability of delay partial difference equations.* — Computers and Mathematics with Applications, Vol. 47, No. 2-3, pp. 345–352.

Toh K. C., Tutuncu R. H. and Todd M. J. (2002): *SDPT3 version 3.02 – a MATLAB software for semidefinite-quadratic-linear programming.* — Department of Mathematics, National University of Singapore, Singapore. Available at http://www.math.nus.edu.sg/∼mattohkc/sdpt3.html

Toker O. and Ozbay H. (1995): *On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback.* — Proc. American Control Conference (ACC), Seattle, USA, 21-23 June, 1995, Vol. 4, pp. 2525–2526.

Toker O. and Ozbay H. (1996): *Complexity issues in robust stability of linear delay-differential systems.* — Mathematics of Control, Signals, and Systems, Vol. 9, pp. 386–400.

Trinh H. and Fernando T. (2000): *Some new stability conditions for two-dimensional difference systems.* — International Journal of Systems Science, Vol. 31, No. 2, pp. 203–211.

Tuan H. D., Apkarian P., Nguyen T. Q. and Narikiyo T. (2002): *Robust mixed H_2/H_∞ filtering of 2-D systems.* — IEEE Transactions on Signal Processing, Vol. 50, No. 7, pp. 1759–1771.

VanAntwerp J. G. and Braatz R. D. (2000): *A Tutorial on linear and bilinear matrix inequalities.* — Journal of Process Control, Vol. 10, No. 4, pp. 363–385.

Vandenberghe L. and Balakrishnan V. (1997): *Algorithms and software for LMI problems in control.* — IEEE Control System Magazine, Vol. 17, No. 5, pp. 89–95.

Vandenberghe L. and Boyd S. (1996): *Semidefinite programming.* — SIAM review, Vol. 38, No. 1, pp. 49–95.

Veres S. M. (2004): *Geometric Bounding Toolbox for MATLAB.* — SysBrain Ltd. Available at http://sysbrain.com/gbt/index.shtml

Vidyasagar M. (1998): *Statistical learning theory and randomized algorithms for control.* — IEEE Control Systems Magazine, Vol. 18, No. 6, pp. 69–85.

Vidyasagar M. and Blondel V. (2001): *Probabilistic solutions to some NP-hard matrix problems.* — Automatica, Vol. 37, No. 9, pp. 1397–1405.

Willems J. C. (1991): *Paradigms and puzzles in the theory of dynamical systems.* — IEEE Transactions on Automatic Control, Vol. 36, No. 3, pp. 259–294.

Wood J., Rogers E. and Owens D. H. (2001): *Behaviours, modules, and duality.* — in K. Gałkowski and J. Wood (eds), Mutidimensional Signals, Circuits and Systems, Systems and Control Book Series, London, England: Taylor and Francis, pp. 45–55.

Wu S.-P. and Boyd S. (1999): *SDPSOL: A parser/solver for semidefinite programs with matrix structure.* — in L. El Ghaoui and S.-I. Niculescu (eds), Recent Advances in LMI Methods for Control, Vol. 2 of Advances in Design and Control, SIAM, pp. 79–91.

Wu S.-P., Vandenberghe L. and Boyd S. (1996): *MAXDET manual.* — Stanford University. Available at http://www.stanford.edu/~boyd/MAXDET.html

Xhafa F. and Navarro G. (1996): *A Maple package for semidefinite programming.* — Research report LSI-96-28-R, Dept. Llenguatges i Sistemes Informatics - Universitat Polit enica de Catalunya. Available at http://www-lsi.upc.es/dept/techreps/ps/R96-28.ps.gz

Xie L., Du C., Soh Y. C. and Zhang C. (2002): *H_∞ and robust control of 2-D systems in FM second model.* — Multidimensional Systems and Signal Processing, Vol. 13, No. 3, pp. 265–287.

Xu S., Lam J., Gałkowski K. and Lin Z. (2004): *An LMI approach to the computation of lower bounds for stability margins of 2D discrete systems.* — Dynamics of Continuous, Discrete and Impulsive Systems, accepted for publication

Xu S., Lam J. and Yang C. (2001): *Quadratic stability and stabilization of uncertain linear discrete-time systems with state delay.* — System and Control Letters, Vol. 43, No. 2, pp. 77–84.

Ye Y., Todd M. J. and Mizuno S. (1994): *An $O(\sqrt{n}L)$-iteration homogeneous and self/dual linear programming algorithm.* — Mathematics of Operations Research, Vol. 19, pp. 53–67.

Youla D. and Gnavi G. (1979): *Notes on n-dimensional system theory.* — IEEE Transactions on Circuits and Systems, Vol. 26, No. 2, pp. 105–111.

Young S. S. (2001): *Stability of discrete state delay systems.* — Proc. 40th IEEE Conf. Decision and Control (CDC), Orlando, USA, 4-7 December, 2001, Vol. 5, pp. 4727–4732.

Zerz E. (2000): *Topics in Multidimensional Linear Systems Theory.* — Lecture Notes in Control and Information Sciences, London, England: Springer-Verlag, Vol. 256.

Zhang B. G. and Deng X. H. (2001): *The stability of certain partial difference equations.* — Computers and Mathematics with Applications, Vol. 42, No. 3-5, pp. 419–425.

Zhou K., Doyle J. C. and Glover K. (1996): *Robust and Optimal Control.* — London, England: Prentice-Hall International.

Lecture Notes in Control and Computer Science

Editor-in-Chief: Józef KORBICZ

Vol. 8: Wojciech Paszke
Analysis and Synthesis of Multidimensional System Classes Using Linear Matrix Inequality Methods
188 p. 2005 [83-89712-81-4]

Vol. 7: Piotr Steć
Segmentation of Colour Video Sequences Using the Fast Marching Method
110 p. 2005 [83-89712-47-4]

Vol. 6: Grzegorz Łabiak
Wykorzystanie hierarchicznego modelu współbieżnego automatu
w projektowaniu sterowników cyfrowych
168 p. 2005 [83-89712-42-3]

Vol. 5: Maciej Patan
Optimal Observation Strategies for Parameter Estimation of Distributed Systems
220 p. 2004 [83-89712-03-2]

Vol. 4: Przemysław Jacewicz
Model Analysis and Synthesis of Complex Physical Systems Using Cellular Automata
134 p. 2003 [83-89321-67-X]

Vol. 3: Agnieszka Węgrzyn
Symboliczna analiza układów sterowania binarnego z wykorzystaniem wybranych metod analizy sieci Petriego
125 p. 2003 [83-89321-54-8]

Vol. 2: Grzegorz Andrzejewski
Programowy model interpretowanej sieci Petriego dla potrzeb projektowania mikrosystemów cyfrowych
109 p. 2003 [83-89321-53-X]

Vol. 1: Marcin Witczak
Identification and Fault Detection of Non-Linear Dynamic Systems
124 p. 2003 [83-88317-65-2]

Prace Naukowe z Automatyki i Informatyki

Przewodniczący: Józef KORBICZ

Vol. 8: Wojciech Paszke
Analysis and Synthesis of Multidimensional System Classes Using Linear Matrix Inequality Methods
188 p. 2005 [83-89712-81-4]

Vol. 7: Piotr Steć
Segmentation of Colour Video Sequences Using the Fast Marching Method
110 p. 2005 [83-89712-47-4]

Tom 6: Grzegorz Łabiak
Wykorzystanie hierarchicznego modelu współbieżnego automatu
w projektowaniu sterowników cyfrowych
168 s. 2005 [83-89712-42-3]

Tom 5: Maciej Patan
Optimal Observation Strategies for Parameter Estimation of Distributed Systems
220 s. 2004 [83-89712-03-2]

Tom 4: Przemysław Jacewicz
Model Analysis and Synthesis of Complex Physical Systems Using Cellular Automata
134 s. 2003 [83-89321-67-X]

Tom 3: Agnieszka Węgrzyn
Symboliczna analiza układów sterowania binarnego z wykorzystaniem wybranych metod analizy sieci Petriego
125 s. 2003 [83-89321-54-8]

Tom 2: Grzegorz Andrzejewski
Programowy model interpretowanej sieci Petriego dla potrzeb projektowania mikrosystemów cyfrowych
109 s. 2003 [83-89321-53-X]

Tom 1: Marcin Witczak
Identification and Fault Detection of Non-Linear Dynamic Systems
124 s. 2003 [83-88317-65-2]