STABILITY ANALYSIS AND SIMULATIONS OF A CLASS OF CONTINUOUS LINEAR 2D SYSTEMS

B. Cichy*, K. Gałkowski*, A. Gramacki**, J. Gramacki**
**Institute of Control and Computation Engineering, **Institute of Computer Engineering and Electronics*
University of Zielona Góra, ul. Podgórna 50, 65-246, Poland

G. Jank
Institute of Mathematics 2
RWTH Aachen, Templergraben 55, D-52062 Aachen, Germany

Keywords: Continuous 2D systems, PDE, LMI.

Abstract: In the paper the problem of stability and stabilization of a class of continuous bidirectional systems has been considered. The system has been described in the form of a continuous linear 2D model. The LMI approach has been successfully applied to the problem. Sufficient conditions of stability has been stated and proofed. Moreover some simulation results have been appended in the last section.

1 INTRODUCTION

Partial differential equations (PDE) play an extremely important role in modelling physical phenomena. They clearly have a strong relationships to 2D systems which are characterized by two directions of the information propagation.

In this paper we consider the class of continuous linear 2D systems described by the partial differential equation of the form

$$\frac{\partial x(t,\tau)}{\partial t} + A_1 \frac{\partial x(t,\tau)}{\partial \tau} = A_2 x(t,\tau) + Bu(t,\tau) \tag{1}$$

with left and right boundary conditions

$$\begin{aligned} x(t,0) &= b_l(t) \\ x(t,\alpha) &= b_r(t) \end{aligned} \tag{2}$$

and initial condition

$$x(0,\tau) = x_0(\tau) \tag{3}$$

where $\tau \in [0,\alpha] \subset \mathbb{R}$ is considered here as the 'space' variable, $t \in \mathbb{R}^+$ is the 'time' variable, $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^r$ is the vector of control inputs. Subscripts l and r state for 'left' and 'right' respectively. Hence the equation (1) defines *initial value* or *Cauchy* problem. If information on x is given at some initial time t_0 (in (3) we assume it as equal to zero) for all τ then the equation (1) describes how $x(t,\tau)$ propagates itself forward in time t. In other words, equation (1) describes time evolution. The goal of numerical codes leading to the solution of (1) with (2) and (3) should be to track that time evolution with some desired accuracy. Notice also that (2) and (3) are examples of *Dirlecht conditions* because they specify the values of the boundary points as (in general) a function of time. Equation (1) is an example of a large class of initial value (time-evolution) PDEs in one space dimension known as a *flux-conservative equations* described in the general form as $\frac{\partial \mathbf{x}}{\partial t} = -\frac{\partial \mathbf{F(x)}}{\partial \tau}$. For our needs we extend it with vector of control inputs u. The illustration of PDE (1) is given on Figure 1 below.

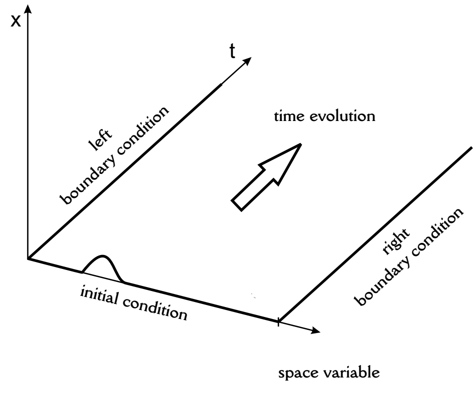

Figure 1: Illustrative scheme of the system (1)

The PDEs (1) are used extensively to model and control chemical processes. For details see (Panagiotis, 2001).

Cichy B., Gałkowski K., Gramacki A., Gramacki J. and Jank G. (2004).
STABILITY ANALYSIS AND SIMULATIONS OF A CLASS OF CONTINUOUS LINEAR 2D SYSTEMS.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 282-287

From the system theory point of view, such a system can be considered as a singular 2D system as there is no the term $\frac{\partial^2 x(t,\tau)}{\partial\tau\partial t}$ in (1). This yields that the standard methods used in such a case for stability analysis and stabilization cannot be used directly and must be reconsidered.

The aim of this paper is to introduce a new physically motivated stability notion, and show preliminary developments in LMI based stability tests and state feedback controller design for this system class. It is to note that LMI techniques are the new, optimization based, very efficient numerically methods, recently extensively used for solving numerous difficult stability related problems, for details see e.g. (S. Boyd and Balakrishnan, 1994). Also a Matlab simulation tool for the processes of this class has been constructed and has been briefly shown.

2 STABILITY ANALYSIS BY USING LMI METHODS

Define Lyapunov function of the form

$$V(x,t,\tau) = x^T(t,\tau)Px(t,\tau) \tag{4}$$

where P is an appropriately dimensioned symmetric positive definite matrix. This enables introducing the following stability definition, which can be easily handled by Lyapunov based techniques.

Definition 1 (Bidirectional stability) *The system of (1) is called bidirectionally stable if and only if is asymptotically stable in both directions t and τ for each possible value of the derivative $\frac{\partial x(t,\tau)}{\partial\tau}$ and $\frac{\partial x(t,\tau)}{\partial t}$ respectively, i.e. there exists such a Lyapunov function of the form (4) that satisfies*

$$\frac{\partial V(x)}{\partial t} < 0, \qquad \forall x(t,\tau),\ \frac{\partial x(t,\tau)}{\partial\tau} \tag{5}$$

$$\frac{\partial V(x)}{\partial\tau} < 0, \qquad \forall x(t,\tau),\ \frac{\partial x(t,\tau)}{\partial t} \tag{6}$$

The following LMI based necessary and sufficient condition for the bidirectional stability can be now developed.

Theorem 1 *The system* (1) *is bidirectionally stable if, and only if, $\exists P_1 > 0$ and $W > 0$, that following LMIs are feasible.*

$$\begin{bmatrix} A_2^T P_1 + P_1 A_2 & -P_1 A_1 \\ -A_1^T P_1 & 0 \end{bmatrix} \leq 0 \tag{7}$$

$$\begin{bmatrix} A_1 W A_2^T + A_2 W A_1^T & -I \\ -I & 0 \end{bmatrix} \leq 0 \tag{8}$$

Proof: First consider the condition (5) and rewrite (1) in the form

$$\frac{\partial x}{\partial t} = -A_1 \frac{\partial x}{\partial\tau} + A_2 x \tag{9}$$

Note that

$$\begin{aligned} \dot V(x) &= \dot x^T P x + x^T P \dot x \\ &= -\frac{\partial x}{\partial\tau}^T A_1^T P x + x^T A_2^T P x \\ &\quad - x^T P A_1 \frac{\partial x}{\partial\tau} \end{aligned} \tag{10}$$

which combined with (9) gives us the requirement that $\exists P_1 > 0$ that following LMI is feasible

$$\begin{bmatrix} A_2^T P_1 + P_1 A_2 & -P_1 A_1 \\ -A_1^T P_1 & 0 \end{bmatrix} \leq 0 \tag{11}$$

In what follows, consider (6). Now (1) can be rewritten as

$$\frac{\partial x}{\partial\tau} = -A_1^{-1}\frac{\partial x}{\partial t} + A_1^{-1}A_2 x \tag{12}$$

and it is straightforward to see that (6) holds if $\exists$ a $P_2 > 0$ that following LMI is feasible

$$\begin{bmatrix} (A_1^{-1}A_2)^T P_2 + P_2(A_1^{-1}A_2) & -P_2 A_1^{-1} \\ -A_1^{-T} P_2 & 0 \end{bmatrix} \leq 0 \tag{13}$$

which can be simplified to the form

$$\begin{bmatrix} A_2^T A_1^{-T} P_2 + P_2 A_1^{-1} A_2 & -P_2 A_1^{-1} \\ -A_1^{-T} P_2 & 0 \end{bmatrix} \leq 0 \tag{14}$$

and left and right multiply (14) by $\mathrm{diag}\{P_2^{-1}, I\}$ to obtain

$$\begin{bmatrix} P_2^{-1} A_2^T A_1^{-T} + A_1^{-1} A_2 P_2^{-1} & -A_1^{-1} \\ -A_1^{-T} & 0 \end{bmatrix} \leq 0 \tag{15}$$

Next left and right multiply this result by $\mathrm{diag}\{A_1, I\}$ and substitute $P_2^{-1} \hat{=} W$ to obtain

$$\begin{bmatrix} A_1 W A_2^T + A_2 W A_1^T & -I \\ I & 0 \end{bmatrix} \leq 0 \tag{16}$$

only if $\exists$ a $W > 0$.

Note however that the test of Theorem 1 cannot be efficiently handled by LMI solvers due to the presence of zero blocks (2,2) in both LMIs and the presence of nonstrict inequalities, which also makes the solution more difficult. Instead, we propose the following approach.

Theorem 2 *The system* (1) *is bidirectionally stable if $\exists P_1 > 0$, $W > 0$ such that the following LMIs are feasible*

$$\begin{bmatrix} A_2^T P_1 + P_1 A_2 & -P_1 A_1 \\ -A_1^T P_1 & -\epsilon I \end{bmatrix} \leq 0 \tag{17}$$

$$\begin{bmatrix} A_1 W A_2^T + A_2 W A_1^T & -I \\ -I & -\epsilon I \end{bmatrix} \leq 0 \tag{18}$$

for $0 < \epsilon \ll 1$.

Proof: Proof is immediate when noting that LMIs of (7)–(8) are the limit cases of (17)–(18) where $\epsilon \to 0$.

The test of the Theorem 2 can be performed efficiently by minimizing $\epsilon > 0$ subject to LMIs (17)–(18). If such an ϵ is close to zero, the process is stable.

3 STATE FEEDBACK STABILIZATION BY USING LMI

Consider a closed loop control law of the standard form

$$u(t,\tau) = Kx(t,\tau) \tag{19}$$

Our aim now is to find a matrix K such that the resulted closed loop system is bidirectionally stable, i.e. $\exists P_1 > 0$ and $W > 0$ such that the following matrix inequalities hold

$$\begin{bmatrix} (A_2+BK)^T P_1 + P_1(A_2+BK) & -P_1A_1 \\ -A_1^T P_1 & -\epsilon I \end{bmatrix} \leq 0 \tag{20}$$

$$\begin{bmatrix} A_1W(A_2+BK)^T + (A_2+BK)WA_1^T & -I \\ -I & -\epsilon I \end{bmatrix} \leq 0 \tag{21}$$

for $0 < \epsilon \ll 1$. However, this result is not in the LMI form and cannot serve as an efficient design tool. But, it is a base for the following main result.

Theorem 3 *The system (1) is bidirectionally stable under the control low of (19) when $\exists W > 0$ and N, such that the following LMIs are feasible*

$$\begin{bmatrix} A_2W + WA_2^T + BN + N^TB^T & -A_1 \\ -A_1^T & -\epsilon I \end{bmatrix} \leq 0 \tag{22}$$

$$\begin{bmatrix} A_1WA_2^T + A_2WA_1^T + A_1N^TB^T + BNA_1^T & -I \\ -I & -\epsilon I \end{bmatrix} \leq 0 \tag{23}$$

for $0 < \epsilon \ll 1$ and then the controller matrix K of (19) is obtained as

$$K = NW^{-1} \tag{24}$$

Proof: First, left and right multiply (20) by $\mathrm{diag}\{P_1^{-1}, I\}$ to obtain

$$\begin{bmatrix} P_1^{-1}(A_2+BK)^T + (A_2+BK)P_1^{-1} & -A_1 \\ -A_1^T & -\epsilon I \end{bmatrix} \leq 0$$

Assume now that $P_1^{-1} \equiv W$ then we have

$$\begin{bmatrix} W(A_2+BK)^T + (A_2+BK)W & -A_1 \\ -A_1^T & -\epsilon I \end{bmatrix} \leq 0$$

Assume also $KW = N$ to obtain (22). Next, substitute $KW = N$ to (21) to obtain immediately (23) what finished our proof. Consider that, stabilization requires common solution W for both LMIs. In (K. Galkowski and Owens, 2002) one can see a similar approach of using LMI technic in control problems.

Consider an unstable example of (1) with following system matrices

$$A_1 = \begin{bmatrix} -0.09 & -0.87 \\ 1.48 & -0.26 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.70 & -0.10 \\ 0.19 & -1.69 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.71 & -0.90 \\ 0.21 & 1.24 \end{bmatrix}$$

The LMIs of Theorem 3 are feasible and the controller matrix K defined by (24) is now

$$K = \begin{bmatrix} 421.8 & 98.16 \\ -174.1 & -268.2 \end{bmatrix}$$

4 NUMERICAL SIMULATIONS

Now we solve numerically and simulate the dynamics of the system of (1) with given boundary conditions (2) and initial condition (3). Hence consider the following model with two states x_1 and x_2 and three inputs u_1, u_2 and u_3 with model matrices as below

$$A_1 = \begin{bmatrix} 0.7 & -0.1 \\ 0.2 & -0.1 \end{bmatrix} \quad A_2 = \begin{bmatrix} -0.5 & 0.6 \\ 0.7 & -1.3 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0.2 & -0.3 \\ 0.1 & 0.2 & 0.3 \end{bmatrix} \tag{25}$$

We consider the solution on the following mesh - 'spatial' direction τ: 60 equally spaced points in $0 \leq \tau \leq 2\pi = \alpha$; 'time' direction t: 40 equally spaced points in $0 \leq t \leq 15$. For illustrative purposes we also show the built-in property of the solver i.e we double the number of spatial mesh points in the second half of the space variable τ.

Initial and boundary conditions must also be supplied. Hence we assume four boundary conditions

$$\begin{aligned} x_1(t,0) &= 0.4 \\ x_1(t,2\pi) &= 0 \\ x_2(t,0) &= -0.2 \\ x_2(t,2\pi) &= 0 \end{aligned} \tag{26}$$

and two initial conditions

$$\begin{aligned} x_1(0,\tau) &= e^{-5(\tau-1)^2} \\ x_2(0,\tau) &= sin(\tau) \end{aligned} \tag{27}$$

We act for the system with three different sets of inputs

Set #1

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{28}$$

Set #2

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{for } \tau > 2$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -0.3 \\ -0.2 \\ -0.1 \end{bmatrix} \qquad \text{for } \tau \leq 2 \tag{29}$$

Set #3 where control values depend on t

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ e^{-t} \\ 0 \end{bmatrix} \tag{30}$$

To obtain a numerical solution, a MATLAB's PDE solver `pdepe` is used - see (MathWorks, 2002) for details. It solves a class of parabolic/eliptic partial differential equation (PDE) systems. The general class to which `pdepe` applies has the form

$$c\left(\tau, t, x, \frac{\partial x}{\partial \tau}\right)\frac{\partial x}{\partial t} = x^{-m}\frac{\partial}{\partial \tau}\left(x^m f\left(\tau, t, x, \frac{\partial x}{\partial \tau}\right)\right) + s\left(\tau, t, x, \frac{\partial x}{\partial \tau}\right) \tag{31}$$

where in our case $0 \leq \tau \leq 2\pi$ and $0 \leq t \leq 15$. The integer $m = 0$ see (MathWorks, 2002) for details. The (in general) function c is a diagonal matrix and the flux and source functions f and s are vector valued. Initial and boundary conditions must be supplied in the following form. For $0 \leq \tau \leq 2\pi$ and $t = 0$ the solution must satisfy $x(t, 0) = x_0(t)$ for a specified function x_0. For $\tau = 0$ and $0 \leq t \leq 15$ the solution must satisfy

$$p_l(\tau, t, x) + q_l(\tau, t) f\left(\tau, t, x, \frac{\partial x}{\partial \tau}\right) = 0 \tag{32}$$

for specified functions p_l and q_l. Similarly for $\tau = 2\pi$ and $0 \leq t \leq 15$,

$$p_r(\tau, t, x) + q_r(\tau, t) f\left(\tau, t, x, \frac{\partial x}{\partial \tau}\right) = 0 \tag{33}$$

must hold for specified functions p_r and q_r. Below there is a source code used in simulation of the example (25) in the form expected by `pdepe` with the third set of control inputs (30).

```
function [x1, x2] = cont2D

m = 0;
alpha = 2*pi;

% values at which the numerical solution is computed
% mesh points
tau = [linspace(0, alpha/2, 20)
       linspace(alpha/2+0.1, alpha, 40)]; % spatial variable
t = linspace(0, 15, 40);                  % time variable

u = [0; 0; 0];
sol = pdepe(m,@pdefun,@pdeic,@pdebc,tau,t,[],u);
x1 = sol(:,:,1)
x2 = sol(:,:,2);

% A surface plot is often a good way to study a solution.
figure;
surf(tau, t, x1);
title('component x_1','FontSize', 12);
xlabel('space variable tau','FontSize', 12);
ylabel('time variable t','FontSize', 12);

figure;
surf(tau, t, x2);
title('component x_2','FontSize', 12);
xlabel('space variable tau','FontSize', 12)
ylabel('time variable t','FontSize', 12);


% --------------------------------------------
function [c,f,s] = pdefun(tau,t,x,DxDtau,u)
c = [1;1];
f = [0;0];
A1 = [0.7, -0.1; 0.2, -0.1];
A2 = [-0.5, 0.6; 0.7, -1.3];
B = [1, 0.2, -0.3; 0.1, 0.2, 0.3];

s = -A1*DxDtau + A2*x + B*[0; exp(-t); 0];

% --------------------------------------------
function u0 = pdeic(tau,u)
u0 = [exp(-5*(tau-1).^2); sin(tau)];

% --------------------------------------------
function [pl,ql,pr,qr] = pdebc(xl,ul,xr,ur,t,u)
pl = [ul(1)-0.4; ul(2)+0.2];
ql = [0; 0];
pr = [ur(1); ur(2)];
qr = [0; 0];
```

The algorithm implemented in `pdepe` is as follow. The routine uses a second-order spatial discretization method based on mesh values of the spatial variable τ. Hence the choice of them has strong influence on the accuracy and cost of computations. The integration in time variable t is performed using `ode15s` solver and hence the timestep is chosen dynamically (MathWorks, 2002). The time t values are used only as points where the solution is returned (and printed) and hence have little impact on the accuracy and cost of computations.

For example to produce Figure 5, Matlab has performed calculations in 305 time points compared with 40 ones chosen by the user for plot the result of calculations. Number 305 has been obtained by analyzing the time variable t inside the function `pdefun()` in

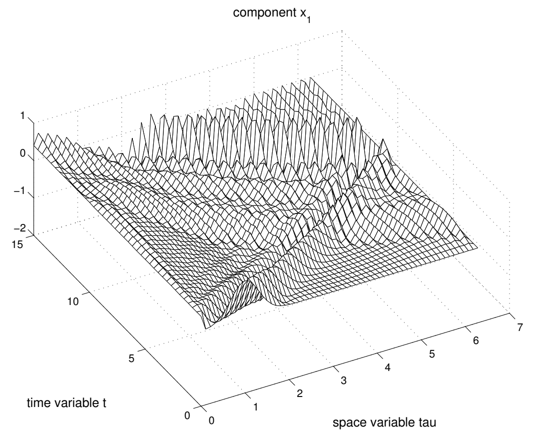

Figure 2: The resulting plot of $x_1(t, \tau)$ with control inputs (29)

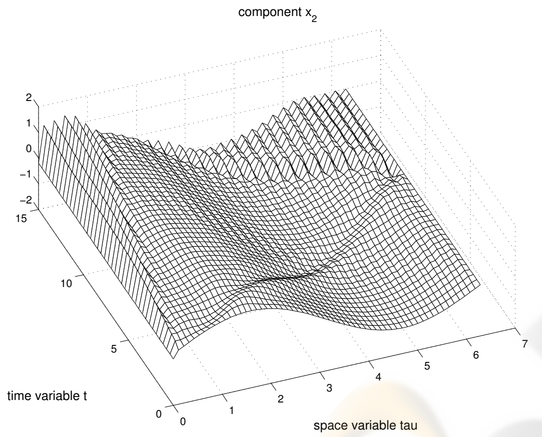

Figure 3: The resulting plot of $x_2(t, \tau)$ with control inputs (29)

the source code above. Figure 4 shows the relationship between 305 calculation points in the time variable and the timestep values chosen dynamically by the solver.

Notice also that the call to `pdepe` above includes the input argument [ ] as a placeholder for additional argument u. The place with [] is reserved for additional options for `ode15s` (see Matlab's help of `odeset` function) which are not used in our case. Moreover the parameter u must be passed to each of the remaining subfunctions even if it is not used there directly.

The resulting plots of $x_1(t, \tau)$ and $x_2(t, \tau)$ with the second set of control inputs (29) i.e. with step control are given on Figures 2 and 3.

The resulting plots of $x_1(t, \tau)$ and $x_2(t, \tau)$ with the third set of control inputs (30) and partially doubled spatial mesh points as described at the beginning of

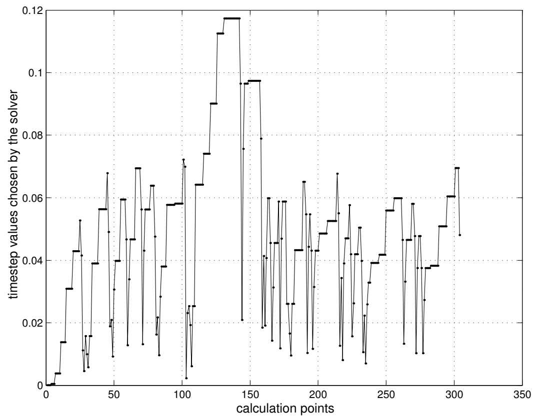

Figure 4: Timestep dynamically chosen by the solver

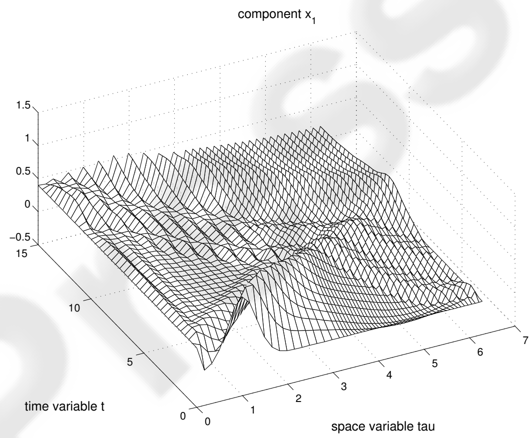

Figure 5: The resulting plot of $x_1(t, \tau)$ with control inputs (30)

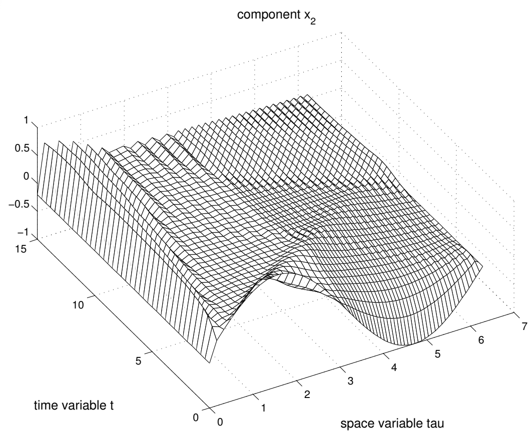

Figure 6: The resulting plot of $x_2(t, \tau)$ with control inputs (30)

this section are given on Figures 5 and 6. The waves, due to constant left and right boundary conditions ”reflects” from this boundaries and finally they turn out even despite of existence of control (it exponentially decrease in time).

5 CONCLUSIONS AND FURTHER WORK

In the paper, the preliminary results on stability and stabilization for a class of singular 2D linear continuous systems has been presented. Also the simulation tool has been developed. Numerical examples show that the stabilisation methods performed here are associated with the comparably high value /energy/ control which can be troublesome. The solution for it can be the guaranteed cost control methods, which is aimed to be the future research topic in this area. Also, the problems with incomplete model information are very important in all practical cases and can be relatively easily solved by LMI techniques.

REFERENCES

K. Galkowski, E. Rogers, S. X. J. L. and Owens, D. H. (2002). *LMIs – a fundamental tool in analysis and controller design for discrete linear repetitive processes*. IEEE Transactions on Circuits and Systems.

MathWorks, T. (2002). *Using MATLAB, Inc. USA*.

Panagiotis, D. P. C. D. (2001). *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport-Reaction Processes*. Birkhauser Boston.

S. Boyd, L. E. Ghaoui, E. F. and Balakrishnan, V. (1994). *Linear Matrix Inequalities in Systems and Control Theory*. SIAM, Philadelphia.