Sérgio Jorge Pereira da Costa

M.Sc.

Multi-Agent Model Predictive Control for Transport Phenomena Processes

Dissertação para obtenção do Grau de
Doutor em Engenharia Química e Bioquímica

Orientadores: José Paulo Barbosa Mota, Prof. Catedrático,
FCT - Universidade Nova de Lisboa
José Manuel P. V. C. Igreja, Prof. Coordenador,
ISEL - IPLisboa

Júri:

Presidente:	Doutora Maria Adelaide de Almeida Pedro de Jesus
Arguentes:	Doutor Nuno Manuel Clemente de Oliveira
	Doutora Carla Isabel Costa Pinheiro
Vogais:	Doutor José Paulo Barbosa Mota
	Doutor João Fernando Pereira Gomes
	Doutor José Manuel Prista do Valle Cardoso Igreja
	Doutor Francisco Avelino da Silva Freitas
	Doutor Mário Fernando José Eusébio

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

May, 2016

Multi-Agent Model Predictive Control for Transport Phenomena Processes

To Maria and Patrícia

Acknowledgements

First of all I would like to thank my two supervisors, Professor José Manuel Igreja and Professor Paulo Mota.

Professor José Igreja for being my friend and my mentor for almost twenty years. My passion on control and mathematical systems came from him and from his enthusiasm. He has been present in every step of my growth as a person and a professional. This work and this thesis is also due to his tenacity and his involvement in every step of the way.

Professor Paulo Mota was the first person with whom I have made contact in Faculdade de Ciência e Tecnologia of Universidade Nova de Lisboa and almost immediately took me under his supervision. Outstanding person, helped me to overcome all obstacles and made everything possible for me to be presenting this work.

A special thanks to Professor João Miranda Lemos from Instituto Superior Técnico, for giving me the opportunity not only to keep on learning about control, but also for giving me permission to use some of his field data on experimental water delivery canal control as a means of validation of my work.

To the Chemical Engineering Departmental Area and Instituto Superior de Engenharia de Lisboa, for providing me the time and opportunity to keep evolving on my personal and institutional enrichment.

I also would like to thank all my colleagues, especially my friend Filipe for encouraging me almost on a daily basis, even when he was finishing his own. A word of appreciation to Valério for his kind words and for his encouragement all the time.

To my parents, my brother and his family for all the love, for all understanding and all the guidance. Without them everything would be much harder and thankfully they were always there for me.

Finally, for the two most important people of my life: my wife Patrícia and my daughter Maria. All the sleepless nights, all the nerves and all the setbacks were overcome with their dedication, their patience and, most of all, their love. Maria was born during the time I have been working on this thesis and I have to thank her for bringing me back to earth and for making me love her every gesture, every word and every care and affection. All of this couldn't have been made without them.

I would also like to dedicate this thesis to two people who are no longer with us: my dear friend Camané (Carlos Candeias) and to Virgílio Igreja (Professor José Igreja's father) who left us last december. May them rest in peace.

Abstract

Throughout the last decades, control systems theory has thrived, promoting new areas of development, especially for chemical and biological process engineering. Production processes are becoming more and more complex and researchers, academics and industry professionals dedicate more time in order to keep up-to-date with the increasing complexity and nonlinearity. Developing control architectures and incorporating novel control techniques as a way to overcome optimization problems is the main focus for all people involved.

Nonlinear Model Predictive Control (NMPC) has been one of the main responses from academia for the exponential growth of process complexity and fast growing scale. Prediction algorithms are the response to manage closed-loop stability and optimize results. Adaptation mechanisms are nowadays seen as a natural extension of prediction methodologies in order to tackle uncertainty in distributed parameter systems (DPS), governed by partial differential equations (PDE). Parameters observers and Lyapunov adaptation laws are also tools for the systems in study.

Stability and stabilization conditions, being implicitly or explicitly incorporated in the NMPC formulation, by means of pointwise min-norm techniques, are also being used and combined as a way to improve control performance, robustness and reduce computational effort or maintain it low, without degrading control action.

With the above assumptions, centralized (or single agent) or decentralized and distributed Model Predictive Control (MPC) architectures (also called multi-agent) have been applied to a series of nonlinear distributed parameters systems with transport phenomena, such as bioreactors, water delivery canals and heat exchangers to show the importance and success of these control techniques.

Keywords: Distributed parameters systems, nonlinear model predictive control, multi-agent control, stability, adaptation, pointwise min-norm control

RESUMO

Ao longo das últimas décadas, a teoria de controlo de sistemas tem prosperado, promovendo novas áreas de investigação e desenvolvimento, em especial no domínio da engenharia química e biológica. O aumento da complexidade e não linearidade nos processos produtivos faz com que investigadores, académicos e profissionais da indústria tenham de dedicar mais tempo ao seu estudo. O desenvolvimento de novas técnicas e arquiteturas de controlo tem ganho preponderância face aos problemas de optimização destes mesmos processos e o foco de atenção para todos os envolvidos.

O controlo preditivo não linear (NMPC) tem sido uma das principais respostas da investigação face ao crescimento exponencial da complexidade dos processos de produção e do aumento de escala dos mesmos. Os algoritmos de predição fazem face à garantia de estabilidade em malha fechada e na optimização dos resultados obtidos. Os mecanismos de adaptação começam a ser vistos como uma extensão natural das metodologias preditivas e são usados para resolver a incerteza associada aos sistemas de pârametros distríbuidos (DPS), compostos por conjuntos de equações diferenciais parciais (PDE). Observadores e mecanismos adaptativos baseados são utilizados para os sistemas aqui apresentados.

Estabilização e condições de estabilidade, implicita ou explicitamente incorporadas na formulação geral do NMPC como controlo do tipo pointwise min-norm, são usadas e combinadas com outras estruturas de forma a melhorar o desempenho, garantir a robustez e manter o esforço computacional baixo.

Tendo como ponto de partida as permissas acima indicadas, foram usadas estruturas de controlo centralizado (ou de agente único) ou controlo descentralizado ou, ainda, distríbuído (controlo multi agente) em sistemas de pârametros distríbuidos com fenómenos de transporte, como sejam bioreactores, canais de distribuição de água ou permutadores de calor por forma a mostrar a importância e o sucesso destas técnicas de controlo.

Palavras-chave: Sistemas de pârametros distribuídos, controlo preditivo não linear, controlo multi-agente, estabilidade, adaptação, controlo pointwise min-norm

CONTENTS

List of Figures

List of Tables

GLOSSARY

CLF Control Lyapunov Function.

cMPC centralized MPC.

DMC Dynamic Matrix Control.

dMPC decentralized MPC.

DMPC distributed MPC.

DPS Distributed Parameter System.

dSIORHC decentralized SIORHC.

FLC Feedback Linearizing Control.

GPC Generalized Predictive Control.

ISS Input-to-State Stability.

LAL Lyapunov Adaptation Law.

MIMO Multiple Input Multiple Output.

MPC Model Predictive Control.

NMPC Nonlinear Model Predictive Control.

OC Orthogonal Collocation.

OCM Orthogonal Collocation method.

ODE Ordinary Differential Equaltion(s).

PDE Partial Differential Equation(s).

PSE Process System Engineering.

PWMN PointWise Min Norm.

PWMNC PointWise Min-Norm Controller.

QP Quadratic Programming.

RCLF Robust Control Lyapunov Function.

RH Receding Horizon.

RHC Receding Horizon Control.

RLCF Robust Lyapunov Control Function.

RPWMN Robust PointWise Min-Norm.

SIORHC Stabilizing Input Output Receding Horizon Control.

SISO Single Input Single Output.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Over the last three to four decades MPC has been extensively studied and developed both in linear and nonlinear systems. MPC can be described as an optimization of the future behaviour of a process by using a model, making the model the basis of the controller. Rawlings presents in [100] an extensive survey focusing on the application of MPC using a very simple and straight forward language to introduce a set of complex terms, as well as presents a typical example regarding the usage of MPC. In [97] the authors focus on industrial application of MPC algorithms provided by vendors, reporting both linear and nonlinear applications, as depicted in 1.1.

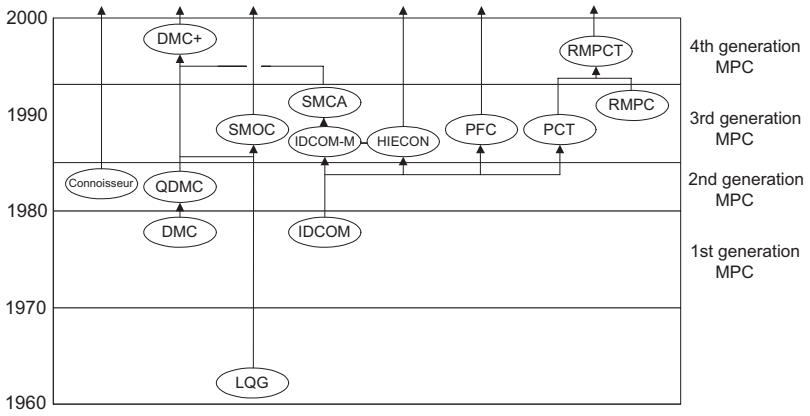

Figure 1.1: Genealogy of MPC algorithms as seen in [97].

The list of papers on the application of MPC in linear systems is massive, but being this thesis about Distributed Parameter System (DPS) in chemical or affine processes, such as tubular bioreactors or water delivery canals, the nonlinearity is present and these

nonlinear systems are far less studied. Take notice that the nonlinearity in these processes is at least promoted by mass and energy conservation laws as well as chemical kinetics and its relationship with the properties of the final products.

1.2 State of the Art

When dealing with DPS several methods have been proposed in order to solve many of the challenges that these systems (especially the nonlinear) introduce. Many researchers and research groups developed mathematical solutions to deal with both linear and nonlinear systems (for instance parabolic or hyperbolic Partial Differential Equation(s) (PDE) systems) has it will be described below.

1.2.1 Operators and Semigroups

In [27] the authors start by motivating what they find has the usefulness of developing a theory for linear infinite dimensional systems and to do so they begin to present a series of control problems to practical and industrial applications. Their next step was to generalize the studied systems in the form:

$$\dot{z} = Az(t) + Bu(t), t \geq 0, \qquad z(0) = z_0 \tag{1.1}$$

where Z is a separable complex Hilbert space. In the example proposed, the authors define the operators A and B as:

$$\begin{aligned} A &= \frac{d^2}{dx^2} \\ D(A) &= \{z \in L_2(0,1); Mz = M_1(.)|_{0,1} + M_2 \frac{d(.)}{dx}|_{0,1} = 0\} \\ B &= I \end{aligned} \tag{1.2}$$

and with $Z = L_2(0,1)$ as the state space, $z(\cdot,t) = \{z(x,t), 0 \leq x \leq 1\}$ as the state, $u(\cdot,t)$ as the input and $z_0(\cdot) \in L_2(0,1)$ as the initial state. The boundary conditions, Mz, are defined in the operator A domain. After the formulation, the solution is obtained and given by:

$$z(x,t) = \int_0^1 g(t,x,y) z_0(y) dy + \int_0^t \int_0^1 g(t-s,x,y) u(y,s) dy ds \tag{1.3}$$

where $g(t,x,y)$ is given by:

$$g(t,x,y) = 1 + \sum_{n=0}^{\infty} 2e^{-n^2\pi^2 t} cos(n\pi x) cos(n\pi y). \tag{1.4}$$

A bounded operator $z(t) = T(t)z_0$ on $L_2(0,1)$ is then introduced in order to simplify the above solution as:

$$z(x,t) = T(t)z_0 + \int_0^t T(t-s)u(s)ds. \tag{1.5}$$

This new bounded operator is what is called a *strongly continuous semigroup* as its properties in $T(t) : \mathbb{R}^+ \rightarrow \mathcal{L}(Z)$ are satisfied:

$$\begin{aligned} T(t+s) &= T(t)T(s), \qquad \text{for} \quad t,s \geq 0; && (1.6)\\ T(0) &= I; && (1.7)\\ \|T(t)z_0 - z_0\| &\rightarrow 0, \qquad \text{as} \quad t \rightarrow 0^+ \qquad \forall z_0 \in Z. && (1.8) \end{aligned}$$

In [32] the authors use a similar deduction for a parabolic PDE with boundary control actuation subject to input and state constraints.

1.2.2 Galerkin Method

One other approach on DPS control was suggested and developed in [11] and further ahead in [24], where the authors primarily describe hyperbolic and/or parabolic distributed systems such as, for instance heat exchangers, and use a different method from the one presented in the previous section in order to solve the set of PDEs. Although there are some differences in the adopted notation in both documents, the DPS can be written as:

$$\frac{\partial v(t)}{\partial t} = Av(t) + Bf(t) \tag{1.9}$$

with the initial conditions $v(t) = v_0$ and output $y(t) = Cv(t)$. The operator A is described like in 1.2.1 and originates a C_0 semigroup $U(t)$, with a growth property:

$$\| U(t) \| \leq Ke^{-\sigma t}, \qquad t \geq 0, \qquad K \geq 1 \quad \text{and} \quad \sigma > 0 \tag{1.10}$$

and B and C operators are defined as:

$$Bf(t) = \sum_{i=1}^{M} b_i f_i(t) \tag{1.11}$$

$$y_j(t) = (c_j, v(t)), \qquad 1 \leq j \leq P. \tag{1.12}$$

with both b_i and c_j being influence functions in the Hilbert space H. This DPS can be written as:

$$v(t) = U(t)v_0 + \int_0^t U(t-\tau)Bf(\tau)d\tau, \qquad y(t) = Cv(t) \tag{1.13}$$

as a weak formulation of the original DPS. A reduced-order model is then obtained:

$$\begin{aligned} \frac{\partial v_N}{\partial t} &= A_N v_N + A_{NR} v_R + B_N f \\ \frac{\partial v_R}{\partial t} &= A_{RN} v_N + A_R v_R + B_R f, \qquad y = C_N v_N + C_R v_R \end{aligned} \tag{1.14}$$

in order to synthesize a finite dimensional feedback controller using the Galerkin method.

In the latter, the author in [24] presents a similar DPS formulation:

$$\begin{aligned} \frac{\partial \bar{x}}{\partial t} &= A\frac{\partial \bar{x}}{\partial z} + B\frac{\partial^2 \bar{x}}{\partial z^2} + wb(z)u + f(\bar{x}) \\ y^i &= \int_\alpha^\beta c^i(z)k\bar{x}dz, \qquad i = 1,\ldots,l \\ q^k &= \int_\alpha^\beta s^k(z)\omega\bar{x}dz, \qquad k = 1,\ldots,p \end{aligned} \tag{1.15}$$

subject to the boundary conditions and initial condition, respectively:

$$\begin{aligned} C_1\bar{x}(\alpha,t) + D_1\frac{\partial \bar{x}}{\partial z}(\alpha,t) &= R_1 \\ C_2\bar{x}(\beta,t) + D_2\frac{\partial \bar{x}}{\partial z}(\beta,t) &= R_2 \end{aligned} \tag{1.16}$$

$$\bar{x}(z,0) = \bar{x}_0(z). \tag{1.17}$$

In this formulation $\bar{x}$ is the state, z and t are respectively space and time, α and β are the domain of the process, u, y and q are, respectively, the manipulated inputs, the controlled outputs and the measured outputs. The functions $b(z)$, $c^i(z)$ and $s^k(z)$ are smooth functions of z. This system can now be written as a infinite dimensional system in Hilbert space H:

$$\begin{aligned} &\dot{x} = \mathcal{A}x + \mathcal{B}u + f(x), \qquad x(0) = x_0 \\ &y = \mathcal{C}x, \quad q = \mathcal{Q}x \end{aligned} \tag{1.18}$$

with:

$$\begin{aligned} \mathcal{A}x &= A\frac{\partial \bar{x}}{\partial z} + B\frac{\partial^2 \bar{x}}{\partial z^2} \\ x \in D(\mathcal{A}) &= \{x \in H([\alpha,\beta];\mathbb{R}^n); \quad C_1\bar{x}(\alpha,t) + D_1\frac{\partial \bar{x}}{\partial z}(\alpha,t) = R_1; \\ &\qquad C_2\bar{x}(\beta,t) + D_2\frac{\partial \bar{x}}{\partial z}(\beta,t) = R_2\} \end{aligned} \tag{1.19}$$

and

$$\begin{aligned} \mathcal{B}u &= wbu \\ \mathcal{C}x &= (c,kx) \\ \mathcal{Q}x &= (s,\omega x). \end{aligned} \tag{1.20}$$

Like in the previous formulations, $\mathcal{A}$ generates a strongly continuous semigroup if the eigenspectrum for $\mathcal{A}$ can be partitioned in finite-dimensional of slow eigenvalues and a stable infinite-dimensional complement of fast eigenvalues and that this separation between slow and fast is large [24]. With so, the general solution for the system can be written in the form:

$$x(t) = U(t)x_0 + \int_0^t U(t-\tau)[(\mathcal{B}u(\tau) + f(x(\tau))]d\tau \tag{1.21}$$

with:

$$\|U(t)\| \leq K_1 e^{a_1 t}, \quad \forall t \geq 0. \tag{1.22}$$

Take notice that the formulations presented are very similar and the authors use the Galerkin approximation method to design the controllers and obtain the solution for these systems.

1.2.3 Orthogonal Collocation Method

As seen so far, the research in DPS applications is based upon different mathematical techniques. In [30] the authors describe a dynamical model for a fixed bed bioreactor in the form:

$$\frac{\partial x_1}{\partial t} = \bar{K_1}\bar{\varphi}(x_1, x_2) \tag{1.23}$$

$$\frac{\partial x_2}{\partial t} = \frac{F}{A}\frac{\partial x_2}{\partial z}\bar{K_2}\bar{\varphi}(x_1, x_2) \tag{1.24}$$

with boundary conditions:

$$x_2(t, z = 0) = x_{2,in}(t). \tag{1.25}$$

In this dynamical model x_1 and x_2 are the states, $\varphi(x_1, x_2)$ is a function in (x_1, x_2) that describe the kinetics, $\bar{K_1}, \bar{K_2}$ and F are, respectively, the yield coefficient matrices and the hydraulic flow rate. Finally, A is the cross-section area of the bioreactor. The authors then propose themselves to achieve a reduced model by using the Orthogonal Collocation method (OCM) [120]. With this method the PDEs are transformed into a series of Ordinary Differential Equaltion(s) (ODE) by expanding the variables as a finite sum of products of time and space functions:

$$x_1(z,t) = \sum_{j=0}^{p+1} B_j(z)x_{1,j}(t), \qquad x_{1,j}(t) = x_{1,j}(t, z = z_j) \tag{1.26}$$

$$x_2(z,t) = \sum_{j=0}^{p+1} B_j(z)x_{2,j}(t), \qquad x_{2,j}(t) = x_{2,j}(t, z = z_j) \tag{1.27}$$

with Lagrange polynomials in the form:

$$\beta_j(z_i) = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases} . \tag{1.28}$$

The reduced model can now be written as a set of ODEs in each collocation point and at the output:

$$\frac{dx_1}{dt} = K_1\varphi(x_1, x_2) \tag{1.29}$$

$$\frac{dx_2}{dt} = -\frac{F}{A}Bx_2 + F_R + K_2\varphi(x_1, x_2) \tag{1.30}$$

where x_1 and x_2 are column vectors of states at each collocation point, $\varphi(x_1, x_2)$ is a column vector of the original $\bar{\varphi}(x_1, x_2)$ also at the collocation points, K_1 and K_2 are diagonal matrices of $\bar{K}_1$ and $\bar{K}_2$, respectively and B_{ij} and F_R are matrices that depend on the derivative of β_{ij} along space at the chosen collocation points (for further understanding please read [30]).

The resulting output will be obtained as:

$$\frac{dY}{dt} = -\frac{F}{A}C^TBx_2 + \frac{F}{A}C_2^T\bar{b}_{p+1}x_{2,in} + \theta^T\phi \tag{1.31}$$

where Y is the value of the controlled component at the reactor output, y_i is the concentration of the controlled component along the reactor at the collocation points z_i, C^TBx_2 is a linear combination of only the variables y_i at the different collocation points and θ and ϕ are column vectors of the unknown and known parameters, respectively. F is the controlled input.

The stated advantages when implementing OCM, for instance when deducing an adaptive controller such as described [30], is that this method is by far easier when compared with the Galerkin method and also that after the model reduction the integrity and nature of the original system remains unchanged. Take note that the mass balances are preserved as demonstrated in [23].

In [50] we propose a similar application as in [30], but for a system modelled as:

$$\frac{\partial w}{\partial t} - \frac{u}{L}\frac{\partial w}{\partial z} = p(w, x, d) \tag{1.32}$$

$$\frac{\partial x_k}{\partial t} + \frac{v_k}{L}\frac{\partial w}{\partial z} = f_k(w, x, d). \tag{1.33}$$

In this case, w is the fluid temperature that circulates countercurrent on a jacket and x_k are the state variables that represent the components for a given set of endothermic reactions in a inner tube (both w and x_k are space and time functions), u, v_k and L are fluid and transport velocities and tube length, respectively.

The application of the OCM is identical to the one described earlier and, in this case, the objective is to control the space distributed output:

$$y(t) = \frac{1}{z_2 - z_1}\int_{z_1}^{z_2} x_p(z, t)dz \tag{1.34}$$

by manipulating $u(t)$ using an adaptive control scheme.

1.2.4 Flat Systems

One other major important mathematical technique for DPS is the concept of flatness and flat systems. Flatness appeared in a series of papers like [37, 38] and later on in [34, 72, 102, 103] and uses the formalism of differential algebra. The authors use this formalism to say that *"a system is said to be flat if one can find a set of variables, called the flat outputs, such that the system is (non-differentially) algebraic over the differential field generated by the set of flat outputs"* [84]. In a more simplistic definition, a system is flat if it exists a set of outputs in the same number as the inputs, such that it is possible to obtain all states and inputs of a system from these outputs without the need for integration. This means that for any given system with states $x \in \mathbb{R}^n$ and inputs $u \in \mathbb{R}^m$ then the system is flat if there are outputs $y \in \mathbb{R}^m$:

$$y = h\left(x, u, \dot{u}, \ddot{u}, \ldots, \overset{(r)}{u}\right) \tag{1.35}$$

such that:

$$x = \varphi\left(y, \dot{y}, \ddot{y}, \ldots, \overset{(q)}{y}\right) \tag{1.36}$$

$$u = \alpha\left(y, \dot{y}, \ddot{y}, \ldots, \overset{(q)}{y}\right). \tag{1.37}$$

A very interesting use of flat systems in chemical engineering is the one published in [65]. In this paper, the authors define the system as a inhomogeneous parabolic DPS in the form:

$$\alpha \frac{\partial x(z,t)}{\partial t} = -\mathcal{A}x(z,t) + u(z,t), \quad z \in [0,1, \quad t > 0] \tag{1.38}$$

subject to the boundary conditions:

$$\mathcal{B}_1 x(0,t) = w_1(t), \quad t > 0 \tag{1.39}$$

$$\mathcal{B}_2 x(1,t) = w_2(t), \quad t > 0 \tag{1.40}$$

and initial conditions and control output, respectively, defined as:

$$x(z,0) = x_0(z), \quad z \in [0,1] \tag{1.41}$$

$$y(t) = x(0,t), \quad t \geq 0. \tag{1.42}$$

This system is described in [27] as a lumped parameter system in state space, although the main difference is that in this formulation $\mathcal{A}$ and $\mathcal{B}$ are described as second and first order differential operators:

$$\mathcal{A}(\cdot) \triangleq \beta_2 \frac{\partial^2}{\partial z^2}(\cdot) + \beta_1 \frac{\partial}{\partial z}(\cdot) + \beta_0(\cdot) \tag{1.43}$$

$$\mathcal{B}_i(\cdot) \triangleq B_{i1} \frac{\partial}{\partial z}(\cdot) + B_{i0}(\cdot), \qquad i = \overline{1,2}. \tag{1.44}$$

The solution is then presented for $x(z,t)$ as:

$$x(z,t) = \sum_{i=0}^{+\infty} a_i(t)\varphi_i(z) = \sum_{i=0}^{+\infty} a_i(t)\frac{z^i}{i!} \tag{1.45}$$

and then approximated, assuming uniform convergence, by its first $N+1$ terms:

$$x(z,t) = \sum_{i=0}^{N} a_i(t)\frac{z^i}{i!}. \tag{1.46}$$

The same approximation can be performed on the control input $u(z,t)$:

$$u(z,t) = u(t)\sum_{i=0}^{m} f_i \frac{z^i}{i!}, \quad f_i \in \mathbb{R}, \quad f_0 \neq 0, \quad m \leq N-2. \tag{1.47}$$

The resulting model 1.38 is now written as:

$$\alpha \sum_{i=0}^{N-2} \dot{a}_i(t)\frac{z^i}{i!} + \beta_2 \sum_{i=0}^{N-2} a_{i+2}(t)\frac{z^i}{i!} + \beta_1 \sum_{i=0}^{N-2} a_{i+1}(t)\frac{z^i}{i!} + \beta_0 \sum_{i=0}^{N-2} a_i(t)\frac{z^i}{i!} = u(t)\sum_{i=0}^{m} f_i \frac{z^i}{i!} \tag{1.48}$$

with boundary conditions 1.39 and output 1.41, respectively:

$$B_{11}a_1(t) + B_{10}a_0(t) = w_1(t) \tag{1.49}$$

$$\sum_{i=0}^{N-1} \frac{1}{i!}\left[B_{21}a_{i+1}(t) + B_{20}a_i(t)\right] = w_2(t) \tag{1.50}$$

$$y(t) = x(0,t) = \sum_{i=0}^{N} a_i(t)\frac{0^i}{i!} \equiv a_0(t). \tag{1.51}$$

This formulation is obtain by using the Power series solution of differential equations.

One other study on flatness and flat systems is presented in [57] where the authors use flat systems theory and motion planning in order to solve an hyperbolic PDE system that describes a distributed collector solar field. Although taken into account the differences between parabolic an hyperbolic systems, the application is similar to the one just described being the differences restricted to notation form.

1.3 Model Predictive Control Architectures

Now, the focus is to understand the main MPC architectures applied in various distributed parameters systems such as bioreactors, water canals and chromatography columns. Understanding the architecture and pointing out the differences and applications leads to a better knowledge on how different architectures were implemented and, in some cases, similar results were attained.

1.3.1 Single Agent or Centralized Control

Centralized MPC (cMPC) as been a widespread control architecture over the last decades in different areas such as chemical and biological engineering [21, 30, 33, 52], power systems [91], supply chains [125], aerospace engineering [63] as well as energy efficiency for construction [123].

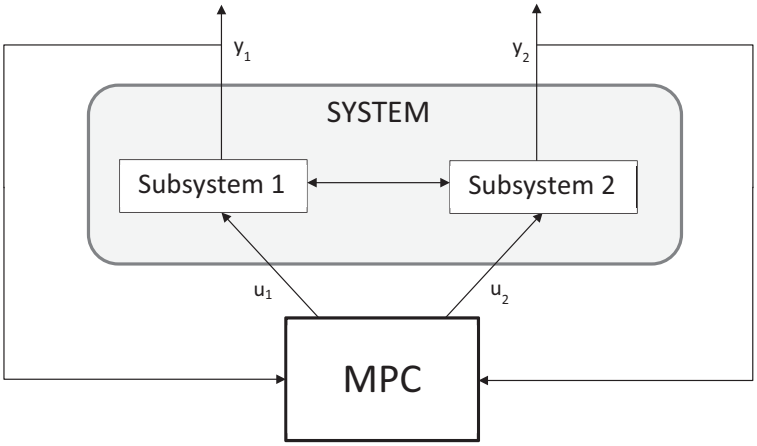

Figure 1.2: Centralized Model Predictive Control architecture.

Commonly applied to linear systems, as for instance in [17, 25], cMPC has suffered compelling advances in nonlinear systems theory and processes. It takes a single multivariable system and its interactions and uses one controller to compute all control actions [105]. By doing so, optimal solutions can be obtained by manipulating the inputs, minimizing the difference between predicted and desired behaviour of a given system or set of subsystems.

Figure 1.2 depicts a generic cMPC applied to a system containing two subsystems. Although of all the advantages in application to linear and low complexity systems, this centralized architecture generates more computer effort as the complexity and nonlinearity of the systems increase.

The evolution of MPC algorithm from a simple unconstrained Single Input Single Output (SISO) case using Dynamic Matrix Control (DMC) and Generalized Predictive Control (GPC) as described in [78] to nonlinear control systems [59, 81] with stability constraints, such as Stabilizing Input Output Receding Horizon Control (SIORHC) [48, 49] and/or with adaptation algorithms [51, 52, 53, 55, 56, 60] is thoroughly discussed in control systems theory literature.

1.3.2 Multi-Agent Control

1.3.2.1 Decentralized Model Predictive Control

The decentralized MPC (dMPC) [10, 18, 121] architecture is still commonly used in large scale industrial systems [98, 121, 122]. As depicted in figure 1.3 different controllers are

used in different subsystems and treated as disjoint sets. These sets of inputs and outputs are coupled in a way that the resulting pairs do not overlap and the local controllers can then be designed to operate independently [105]. The decentralized framework has as main aspect the fact that the controllers inside a given system do not communicate with each other, which means that there is no information trading, although as seen in [47] information is exchanged only for special purposes.

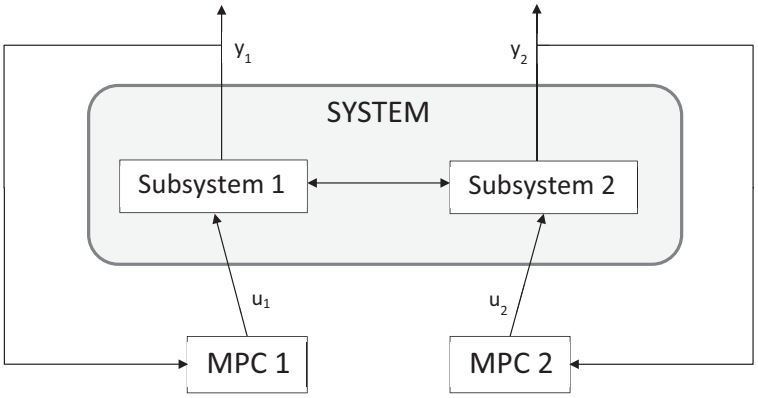

Figure 1.3: Decentralized Model Predictive Control architecture.

In terms of stability analysis a great number of papers have been presented [6, 80, 116, 117, 124], giving particular relevance to the application of Input-to-State Stability (ISS) which was first introduced in [110] and further developed in [62, 98, 112] on nonlinear discrete-time systems or for constrained uncertain nonlinear systems [73].

1.3.2.2 Distributed Model Predictive Control

Contrariwise to dMPC, on distributed MPC (DMPC) [35] information is exchanged between the controllers in order to have coordination in their actions. One of the major classes of DMPC are non-cooperative [75, 76] where the communication between controllers can be one-directional and the controllers are evaluated in a sequence, Sequential DMPC, or at the same time, Parallel DMPC. The other class is cooperative DMPC [39, 69, 77, 90, 113] where the same global cost function is optimized in each controller. In the last decades several research developments have been achieved with the use of DMPC, namely advances in thermal and energy efficiency [12, 13, 14, 15] and for chemical engineering systems [87, 108, 115]. Recent design methods were developed to solve two major problems: guaranteeing stability and improving performance (please read [42, 64, 118]).

1.3.2.3 Non-cooperative DMPC

Like was said before, non-cooperative DMPC uses a set of local controllers whose actions are taken in order to optimize local cost functions. With this type of configuration there is no need for a given subsystem to know what happens, trajectory-wise in the other subsystems, including its neighbours. The exception is mostly focused in guaranteeing

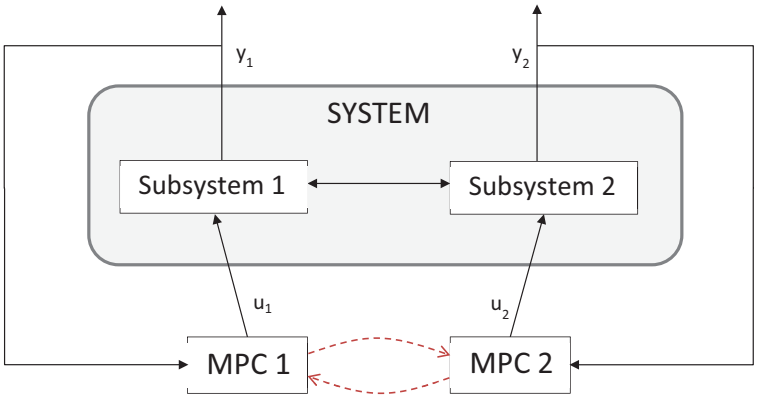

Figure 1.4: Distributed Model Predictive Control architecture.

stability. In order to obtain some solution, at each sampling time, these controllers can exchange information only once (non-iterative parallel controllers) or can iterate (iterative parallel controllers). The iterative algorithms deal with a higher amount of information being exchanged, rather than non-iterative algorithms. Sequential and parallel designs are depicted in figures 1.5 and 1.6 respectively. Mathematical background for non-cooperative game theory can be read in [8].

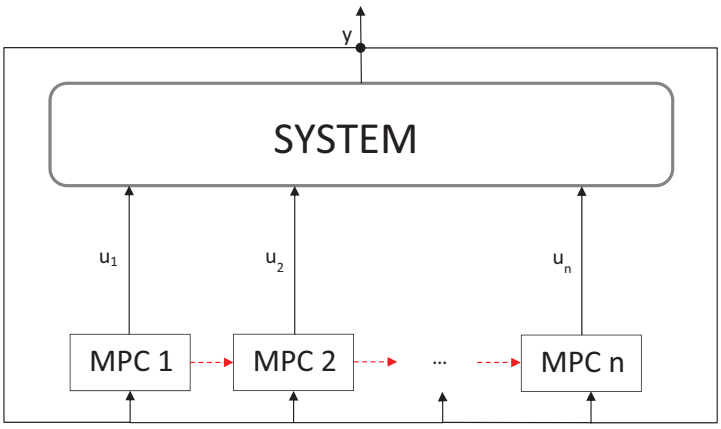

Figure 1.5: Sequential Distributed Model Predictive Control architecture.

1.3.2.4 Cooperative DMPC

In terms of cooperative DMPC the main objective is the optimization of a global cost function, like in a centralized architecture. This means that, although there is the manipulation of local variables in each subsystem (like in a decentralized or non-cooperative architectures) all these manipulations take into account a global optimal solution - Pareto optimal solution - rather than a typical Nash equilibrium of non-cooperative designs. So, the importance of cooperative DMPC is focused on performance improvement by using distributed optimization algorithms. In other words, cooperative architectures operates in a decentralized away, using centralized assumptions. Stability in cooperative DMPC

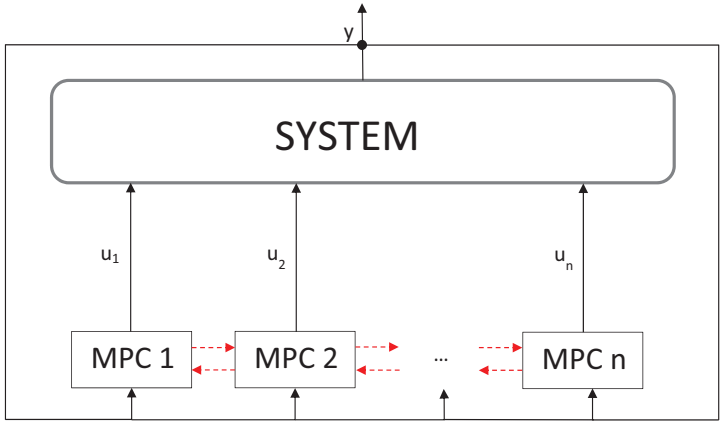

Figure 1.6: Parallel Distributed Model Predictive Control architecture.

should be taken into account in suboptimal control theory. Interesting applications can be seen in [79].

1.4 Thesis structure

This thesis is, therefore, organized as follows:

- Chapter 1 includes a summary of the whole thesis, namely reports the main motivation for the writing of this thesis, the state of the art in terms of the proposed work, different MPC architectures applied to nonlinear systems, including a brief description of centralized, decentralized and distributed designs to be further developed and the organization of all chapters.
- Chapter 2 defines a general formulation for the class of uncertain nonlinear systems with transport phenomena proposed, introduces stability constrainment and applies receding horizon control with adaptation for a fixed-bed tubular bioreactor as an application of the given system.
- Chapter 3 extends the subjects presented in chapter 3 to other systems within the same class of distributed and uncertain nonlinear systems, namely to a finite escape traveling time distributed system. An application to a water delivery canal is thoroughly presented and discussed in SISO and Multiple Input Multiple Output (MIMO) cases with different control architectures, namely centralized and decentralized control.
- Chapter 4 is the extension of the previous chapter with the introduction of cooperative distributed control (multi-agent control) with zero state terminal constraints. New control techniques, mainly neighbour-to-neighbour cooperation have been applied to the water delivery canal prototype, as well as the developed generic serially chained systems solution.

- Chapter 5 makes use of assumptions and definitions from previous chapters and uses them as an application for a countercurrent heat exchanger. Centralized and non-cooperative distributed control techniques are developed and compared in terms of performance, robustness and computational effort.
- Chapter 6 summarizes conclusions for the whole thesis and presents future and under development work.

CHAPTER 2

NONLINEAR MODEL PREDICTIVE CONTROL IN BIOREACTORS

2.1 Introduction

In this chapter, the main goal is, first of all, to introduce the importance of NMPC in control systems theory and invoke some of the work that has been developed for this thesis as an application to a fixed bed tubular bioreactor.

Previously presented and developed for linear systems, in [36] and later on in [44, 99] NMPC has been proven to show important advantages in terms of advanced process control. This is due to the fact that constraint incorporation tends to be easier, becoming a decisive advantage for industrial applications when compared to other methods. The extension to multivariable cases can also be easier and well understood.

However, from an implementation standpoint, a major issue and main focus on research consists in being able to ensure stability for a short finite horizon in an uncertain environment, without dramatically increasing the computational effort. Usually, when in presence of uncertainty in a already complex nonlinear system, closed-loop stability has been shown to be extremely hard to achieve. The solution for these problems is the use of larger weight penalty on control action included in the optimization problem or bigger finite horizon in the controller, which tends to transform the optimization into higher computational effort. From a practical perspective, closed-loop stability is not decisive, and therefore many MPC techniques applied in process industries do not meet this feature. However, as can be read in [107] stability is convenient.

Later on in this chapter a stabilizing NMPC for a given class of uncertain biosystems with transport phenomena, associated with fluid flow in pipes, is to be formulated. The inclusion of a stabilizing condition based in a Robust Lyapunov Control Function (RLCF) (also known as PointWise Min-Norm Controller (PWMNC) as described in [40, 95, 126] and

to be developed in this chapter and further on in chapter 3) will guarantee robust NMPC closed-loop stability. A stable distributed parameter estimator, in the Lyapunov sense, is also to be presented when combined with the NMPC algorithm, using the certainty equivalence principle for adaptation proposes.

Notwithstanding the main attention in the last decade is still on NMPC as in both academic and basic research as well as industrial applications, one of the still less approached area is the incorporation of adaptation mechanisms as mentioned in [85], which seems to be a natural idea to deal with parametric uncertainty (maintaining, of course, low computational cost/effort).

The literature (mostly academic papers such as [3, 5]) on the subject are now gaining some more relevance and meaning. Early results for distributed systems can be found in [56]. Predictive control of hyperbolic PDE systems, namely transport-reaction processes, was studied in [33, 106] for SISO cases. In [33] the controller is based on a predictive model developed using the method of characteristics (for this and other methods and definitions in both hyperbolic and parabolic PDE system solving, please read the seminal [26]) and does not consider constraints. For [106] authors use finite differences method for space discretization and a space distributed actuator was used with success. In [53] the authors used a combination of adaptive and predictive control that was obtained via Orthogonal Collocation (OC) [101] reduced modelling, also for SISO hyperbolic tubular transport-bioreaction processes, that proved to achieve the control objectives. Stability conditions have also been derived and discussed in the same article. A pioneer research on adaptive control of tubular bioreactors, using OC reduced models and Feedback Linearizing Control (FLC) can be found in [30] and also a successful application for a distributed uncertain solar power plant, where NMPC was combined with FLC and Lyapunov Adaptation can be read in [54].

The main contribution of this chapter consists in the formulation of a Lyapunov stable efficient adaptive NMPC for a broad class of parametric uncertain nonlinear PDE systems, that exhibits (bio)mass and energy transport. Therein a stable distributed law, suitable for the entire system class, is used to achieve adaptation, requiring low computational effort. Stability is ensured online by a constraint justified via a Robust Control Lyapunov Function (RCLF) condition that arises from the relation between PWMNC and RHC.

This chapter is organized as follows: after this introduction, the prototype model of the tubular biosystem class is considered and described by a set of PDEs. Afterwards, the general stabilizing formulation of a NMPC for the infinite dimension system class is introduced and then the stability condition is derived. Afterwards, a distributed adaptive law in the sense of Lyapunov (Lyapunov Adaptation Law (LAL)) is obtained and the adaptive NMPC is stated by combining both Receding Horizon (RH) and LAL. Finally, an application to a fixed-bed tubular bioreactor (with Contois kinetics) is described in a series of simulations.

2.2 Class of PDE models

Start by considering the dynamical model described as:

$$\frac{\partial x(z,t)}{\partial t} + \mathcal{L}(x(z,t), u(t); \theta) = s(x(z,t), u(t); \theta), \tag{2.1}$$

where space z and time t have the domain $(z,t) \in [0,1] \times \mathbb{R}^+$, the state trajectories $x(.,t)$ defined as $x(.,t) \in \mathbb{X} \subset [0,1] \times \mathbb{R}^n$, bounded manipulated input defined as $u(t) \in \mathbb{U} \subset \mathbb{R}^m$ and $\mathcal{L}(.,u;\theta)$ as quasi-linear matrix space operator. The boundary conditions are given by the nonlinear space operator $\mathcal{M}(.,u;\theta)$:

$$\mathcal{M}(x(z,t), u; \theta) = 0 \tag{2.2}$$

and the output defined as $y(t) \in \mathbb{Y} \subset \mathbb{R}^p$ is given by:

$$y_k(t) = \int_0^1 b_k(z)\ h_k(x(z,t))dz, \qquad (k = 1, ..., p). \tag{2.3}$$

Both $s(x,u)$, $h(x)$ are smooth vectors of nonlinear functions, defined as $s(x,u) : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$, $h(x) : \mathbb{R}^n \mapsto \mathbb{R}$. The space weight $b_k(z) > 0 : [0,1] \to \mathbb{R}^+$ satisfies $\int_0^1 b_k(z)\ dz = 1$. Finally, uncertain parameters or additive disturbances defined as $\theta \in \Theta \subset \mathbb{R}^q$. This uncertain parameters or additive disturbances lie within a known open convex set $\Theta = \{\theta : \underline{\theta} < \theta < \overline{\theta}\}$.

It is important to declare that the $\mathcal{L}$ operator must include the convection terms $v\frac{\partial(.)}{\partial z}$ and that it will depend on the manipulated variable when $u \equiv v$ (where v is the fluid velocity). If the manipulated variable is space weighted additive in the production term, then $s(x,u;\theta) = s_x(x;\theta) + s_u(x;\theta)\ w(z)\ u$. In both cases the manipulated variable is explicit on equation (2.1) and it will be implicit when it only appears in the boundary condition (2.2).

As stated before this prototype class allows the study of a wide variety of processes with transport phenomena and defines the class of systems presented above. Take note that this set of equations promotes a process model that occurs in a tubular or cylindrical domain. Here the state variables $x_k(z,t)$ $(k = 1, \ldots, n)$ represent the (bio)chemical species involved and the mixture temperature along a normalized space $(z \in [0,1])$ and time $(t \in [0, +\infty[)$, where v is the transport velocity. The system is thereby modelled by n quasi-linear PDEs in the state variables $x_k(x,t)$, that results from ensuring the application of conservation principles (mass or biomass and energy balances) as described in [7, 20, 31].

2.3 Process System Engineering design assumptions

The class of systems just presented arises in Process System Engineering (PSE) and are designed to operate in steady state for long periods of time. A change in operating conditions is usually promoted by a sudden set-point jump between two steady states nearby.

Keeping this in mind there are some assumptions that need to be made.

Assumption A1: By design, there exist one or more steady state nominal operating points and for each nominal operating point there exists a output value y_r, a corresponding state profile $x_r^0 \equiv x_r(z;\theta)|_{\theta=\theta_0}$ and input $u_r^0 \equiv u_r(\theta)|_{\theta=\theta_0}$ obtained for the nominal parameter values $\theta = \theta_0 \in \Theta$. And also a subset $\mathbb{X}_r(\theta_0)$ given by:

$$\mathbb{X}_r(\theta_0) = \{x \in \mathbb{X} :\parallel x - x_r^0 \parallel_{Q_r} \leq l_r\} \tag{2.4}$$

$$l_r = \max_{\theta\in\Theta}\{\parallel x_r(z;\theta) - x_r^0 \parallel_{Q_r}\}, \tag{2.5}$$

where $\mathbb{X}_r$ is a closed convex subset, containing x_r and x_r^0, bounded by the curve level l_r of the elliptical hyper paraboloid $V_r(x) = \parallel x - x_r^0 \parallel_{Q_r}$.

Take notice that the actual steady state space profile $x_r(z;\theta)$ does not coincide with the nominal state space profile, namely because in general $\theta \neq \theta_0$ due to parametric uncertainty, and by so not accurately known but lying inside $\mathbb{X}_r$ ($x_r(z;\theta) \in \mathbb{X}_r(\theta_0)$). Figure 2.1 depicts what is described in *Assumption A1*.

Assumption A2: Properness *viz.* $\mathbb{X}_r \subsetneq \mathbb{X}$ and so by *A1*, x_r and $x_r^0 \in \mathbb{X}_r \subset \mathbb{X}$.

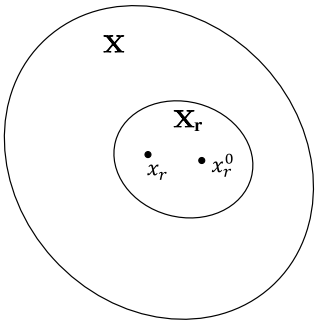

Figure 2.1: Sets.

2.4 Stability

In terms of stability, as shown in [95] one can develop a PointWise Min Norm (PWMN) control based in a RLCF as defined in [40] for the stated class of systems. As for the NMPC problem previously stated it is suggested that, at the same time, it is possible to include the PWMN stabilizing property as an inequality constraint in its basic formulation.

In [40] the authors declare that an implicit stabilizing controller may be design using the following optimization statement: *if* $V(\tilde{x}) > l_0$ *then*:

$$\begin{aligned} &\min_{u\in\mathbb{U}} \quad u^T u \\ &s.\,t.\ (2.1)\ and\ \max_{\theta\in\Theta}\{\dot{V}(\tilde{x};\theta)\} + \alpha V(\tilde{x}) < 0, \end{aligned} \tag{2.6}$$

where $V(\tilde{x}) : X \to \mathbb{R}_{\geq 0}$ is a continuously differentiable, positive definite and radially unbounded function in respect to the L_2 norm of $\tilde{x}$, as seen in [27]. The difference between the actual state and a steady state profile along space length $\tilde{x}(z,t) = x(z,t) - x_r(z;\theta_0)$, where $x_r \in \mathbb{X}_r$, obtained for parameter nominal value θ_0. In other words, $V(\tilde{x})$ is simply a RLCF candidate whose maximum derivative can be made less than $-\alpha V(\tilde{x})$ pointwise (see [41]) by the choice of control values outside a region or set around the nominal operating point. The region must be chosen by adjusting the curve level value l_0 and can converged using the α relaxation parameter.

In terms of choice for the Control Lyapunov Function (CLF) candidate:

$$V(e) = \frac{1}{2}\int_0^1 \tilde{x}^T q(z)\tilde{x}\, dz, \qquad \int_0^1 q(z)\, dz = 1 \tag{2.7}$$

with $q(z)$ positive definite. Using Lyapunov stability arguments, as shown in [67] and in[86]: any $\tilde{x}(z,t)$ solution, originating in a bounded region, will asymptotically tend to the included invariant region parameterized by l_0 as $t \to \infty$, if $\dot{V} < 0$ $(\forall\ \tilde{x} \neq 0)$. In this case, the time derivative of (2.7) will yield:

$$\dot{V} = \frac{1}{2}\int_0^1 \frac{\partial}{\partial t}\left(\tilde{x}^T(z,t)\ q(z)\ \tilde{x}(z,t)\right)\ dz \tag{2.8}$$

using (2.1):

$$\dot{V} = \int_0^1 (s(x,u;\theta) - \mathcal{L}(x,u;\theta)^T q(z)\tilde{x}(z,t)\ dz \tag{2.9}$$

and the robust optimization condition, for this class of systems, is given by:

$$\max_{\theta}\left\{\int_0^1 (s(x,u;\theta) - \mathcal{L}(x,u;\theta))^T q(z)\tilde{x}(z,t)\ dz\right\} + \frac{\alpha}{2}\int_1^0 \tilde{x}^T q(z)\tilde{x}\, dz < 0. \tag{2.10}$$

Note that one of the following conditions must hold *a priori* in relation to the manner how the inputs appear in (2.1). Condition $\int_0^1 (\frac{\partial x}{\partial z})^T A_v\ q(z)\tilde{x}\, dz \neq 0$, where A_v is a diagonal matrix with one or zero in the main diagonal, must hold if the corresponding state is related with the manipulated velocity. Or

$$\int_0^1 (s_u(x;\theta)\ w(z)\ u)^T q(z)\tilde{x}\, dz \neq 0$$

iff $u \neq 0$, if u is related with the production term. Finally, $\mathcal{M}(x,u,\theta) \neq 0$ *iff* $u \neq 0$ if u is implicit through boundary conditions. These conditions must hold or controllability from u to the 'output' $\equiv V$ is lost. Note also that in order for the RCLF to be chosen the conditions on $q(z)$ must be able to find V and the corrected condition must hold. If this does not happen, then the candidate V is not a RLCF and must be discarded and one must find a new CLF candidate that can verify all conditions. If nothing works then other methods must be applied.

The established above result can be stated in the following proposition:

Proposition 1: Consider the class of distributed systems $\Sigma_\Delta = (\mathcal{L}, \mathcal{M}, s, \mathbb{U}, \mathbb{X}, \mathbb{X}_r, \mathbb{Y}; \Theta)$ with solutions $x(z,t)$ defined by (2.1), then the function V given by (2.7) is a RLCF for Σ_Δ if and only if exists scalars $l_0,\ \alpha \in \mathbb{R}^+$ such that:

$$\min_{u\in\mathbb{U}} \max_{\theta\in\Theta} \{\dot{V}(\tilde{x};\theta)\} + \alpha V(\tilde{x}) < 0$$

whenever $V(\tilde{x}) > l_0$.

Remark 1: The $u \in \mathbb{U}$ referred in proposition 1 can be obtained by the optimization statement (2.6) when feasible, which means that the set $\mathbb{U} \ni u$ must be chosen large enough. Feasibility in the last proposition arises only from the fact that in general unstable systems cannot be stabilized globally when input constrains are present. Without feasibility there is no guarantee for global stability.

Remark 2: An important PWMN control feature is the fact that it corresponds to a NMPC limit stabilizing solution when the horizon value goes to zero. Consider the following NMPC formulation:

$$\begin{aligned} \min_{u\in U} \quad & \int_0^{\mathcal{T}} (V(\tilde{x}(z,\tau)) + \|\tilde{u}\|_R^2) d\tau \\ s.\,t.\ (2.1) \text{ and } & \max_{\theta\in\Theta} \{\dot{V}(\tilde{x};\theta)\} + \alpha V(\tilde{x}) < 0, \end{aligned} \tag{2.11}$$

when the horizon $\mathcal{T}$ tends to zero:

$$\begin{aligned} & \min_u \lim_{\mathcal{T}\to 0} \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} (V(\tilde{x}(z,\tau)) + \|\tilde{u}\|_R^2) d\tau \\ = & \min_u \{V(\tilde{x}(z,t)) + \|\tilde{u}\|_R^2\} \Leftrightarrow \min_u \|\tilde{u}\|_R^2, \end{aligned} \tag{2.12}$$

showing the equivalency to the PWMN control. Make note that dividing (2.11) by $\mathcal{T}$ has no effect on the optimization problem and also that when $\mathcal{T}$ goes to zero there is no need to include the term $V(\tilde{x})$ because it is not affected by u and, so forth, constant. Hence this simple observation, stated in [95], indicates that as $\mathcal{T}$ goes to zero, RH controllers loses the ability to maintain acceptable performance by just minimizing input energy. This implies a degradation in the performance of the controller which leads to closed-loop instability if the robust PWMN condition or some other equivalent mechanism is not included. Make note also that the constraint requires V to be a RCLF for any receding horizon value in order to ensure closed-loop stability. It is suggested further reading on this subject in [52] to obtain extra details about the robust stability conditions.

Remark 3: Due to uncertainty $l_0 \geq l_r$ assuring that any solution $x(z,t)$ will enter and stay inside a set containing $\mathbb{X}_r$ in finite time.

2.5 Proposed NMPC general formulation

The aim is to control the output $y(t)$ in equation 2.3, a state nonlinear function weighted in the space domain, by manipulating the input $u(t)$. The proposed way to achieve this is

by solving an open loop optimization problem and applying a receding horizon strategy according to the NMPC approach like the one presented in [44, 99]. Therefore, defining the basic optimal control problem with quadratic cost functional:

$$\min_{u} J = \int_{t}^{t+\mathcal{T}} \left(\|\tilde{y}(\tau)\|_{Q}^{2} + \|\tilde{u}(\tau)\|_{R}^{2} \right) d\tau, \tag{2.13}$$

where $Q \geq 0$ and $R > 0$ are weighting matrices, subject to the model (2.1)-(2.3) with operational constraints of the general form:

$$\mathcal{O}(\xi(x(t)), y(t), u(t), t) \leq 0 \tag{2.14}$$

and guaranteed closed-loop stability constraint:

$$\max_{\theta \in \Theta} \{ \dot{V}(\tilde{x}; \theta) \} + \alpha V(\tilde{x}) < 0. \tag{2.15}$$

In equations (2.13) to (2.15):

$$\begin{aligned} V(\tilde{x}) &= \frac{1}{2} \int_{0}^{1} \tilde{x}^{T} q(z) \tilde{x} \, dz \\ \xi(x(t)) &= \int_{0}^{1} B(z) \zeta(x) \, dz, \quad \int_{0}^{1} B(z) \, dz = I, \\ \tilde{y} &= y_r - y, \quad \tilde{u} = u_r - u \ \text{ and } \ \tilde{x} = x_r - x, \end{aligned} \tag{2.16}$$

where y_r, x_r and u_r define the reference trajectory to track, $\xi(t)$ is broad set of state functions and $q(z)$ is a space weighting matrix. The stability condition included forces the RHC to have a performance that is equal or higher to a stabilizing control law when applied in the same conditions (see [53]). One approximated computationally efficient procedure for solving the stated nonlinear, infinite dimension, non-convex programming problem is to use a finite parametrization for the control signal. This way, $u(t) \in [t, t + T[$, where N_u segments of constant value $u_1, \ldots, u_{N_u}$ and duration $\frac{T}{N_u}$, as decision variables. Thus the suboptimal, finite dimension, constrained programming problem amounts to solve:

$$\begin{aligned} \min_{u(\bar{t})} J &= \int_{t}^{t+\mathcal{T}} \left(\left(\|\tilde{y}\|_{Q}^{2} + \|\tilde{u}\|_{R}^{2} \right) \right) d\tau \\ s.\,t. \quad & \mathcal{O}(\xi(x(\bar{t})), y(\bar{t}), u(\bar{t}), \bar{t}) \leq 0 \\ & \max_{\theta \in \Theta} \{ \dot{V}(\tilde{x}(t); \theta) \} + \alpha V(\tilde{x}(t)) < 0 \\ & u(\bar{t}) = seq\{u_1, \ldots, u_{N_u}\} \end{aligned} \tag{2.17}$$

and also subject to the proper space semi-discrete model obtained from the original distributed one reposted in [53], where $u(\bar{t})$ is the sequence of steps of amplitude u_i and the variable $\bar{t}$ represents virtual time during the minimization computation, $\bar{t} \in [0, \mathcal{T}[$. Once the minimization result $u(\bar{t})$ is achieved, the first sample u_1 is therefore applied at $t + \delta$ and the whole procedure is repeated. The interval δ corresponds to the time needed to obtain a solution, although is must be assumed that δ must be much smaller than the

sampling interval. Note that if θ is uncertain, there is the need to use an estimate $\hat{\theta}$ in order to perform the minimization (this can be seen in [51]). Also, as stated in [99] the MPC solution (2.17) "*is best regarded as a practical means of implementing the Dynamic Programming solution (control law)*" of (2.13-2.16).

2.6 Adaptive Control

The usage of observers in the estimation of parameters seems to be a good way to tackle parameter uncertainty as long as the state is considered accessible [3, 56]. With this, consider the model (2.1) in plug-flow conditions and with the manipulated variable defined as the fluid velocity:

$$\mathcal{L}_x = \mathcal{L}(x(z,t), u(t); \theta) = \frac{u}{L} A_v \frac{\partial x(z,t)}{\partial z}, \tag{2.18}$$

where A_v is a diagonal matrix that was one or zero main diagonal elements, relating the existence or non-existence of the transport term. Considering that typical process production terms are, in many cases, linear affine in θ one can assumed that:

$$s(x;\theta) = s_0(x) + \sum_{i=1}^{q} \theta_i s_i(x) = s_0(x) + S(x)\theta, \tag{2.19}$$

where $s_0(x)$ is, as stated, a vector of smooth nonlinear functions $(n \times 1)$, $S(x)$ is also a matrix of smooth nonlinear functions $(n \times q)$ and $\theta = [\theta_1 \cdots \theta_q]^T$ is a vector of uncertain parameters assumed to be time constant or, at most, very slowly varying. The observer filter dynamics as seen in [3, 114] takes the form:

$$\frac{\partial x_a(z,t)}{\partial t} + \frac{u}{L} A_v \frac{\partial x(z,t)}{\partial z} = s(x;\hat{\theta}) + \mathcal{K}(x - x_a), \tag{2.20}$$

where $\mathcal{K} > 0$ and the observation error dynamics ($e_a = x - x_a$) is given by:

$$\frac{\partial e_a}{\partial t} = S(x)\tilde{\theta} - \mathcal{K} e_a$$

with $\tilde{\theta} = \theta - \hat{\theta}$. Finally, introducing the Lyapunov candidate function [67]:

$$V(e_a, \tilde{\theta}) = \frac{1}{2}\left(\int_0^1 e_a^T e_a \, dz + \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \right), \tag{2.21}$$

where Γ is a weighting matrix, differentiating V with respect to time, using the error dynamics and choosing:

$$\dot{\hat{\theta}} = -\Gamma \int_0^1 S(x)^T e_a \, dz \tag{2.22}$$

it guarantees $\dot{V}(e_a) < 0$ when $e_a \neq 0$.

This way the errors tends to zero ($e_a(t)$ and $\tilde{\theta} \to 0$ as $t \to \infty$), after initial conditions transient ($w(0)$, $x(0)$ and $\hat{\theta}(0)$), if the nonlinear functions $S(x) \neq 0$ locally. Note that $\tilde{\theta}$ can

be bounded by the use of the projection method as described in [66]:

$$\text{Proj}\left\{\tau,(\overline{\theta},\underline{\theta},\epsilon)\right\} = \begin{cases} max(0,\frac{\epsilon-\hat{\theta}+\overline{\theta}}{\epsilon})\tau & \hat{\theta}\geq\overline{\theta}\ and\ \tau>0 \\ max(0,\frac{\epsilon+\hat{\theta}-\underline{\theta}}{\epsilon})\tau & \hat{\theta}\leq\underline{\theta}\ and\ \tau<0 \\ \tau & otherwise \end{cases} \tag{2.23}$$

with $\overline{\theta}+\epsilon\geq\hat{\theta}\geq\underline{\theta}-\epsilon$ and where τ is the update law and, thereby, constraining the estimated to the interior of a bounded convex parameters space.

When combining NMPC with LAL dynamics, the RHC+LAL solution is formulated, like in [51], using (2.13) subject to the predictive model (2.1), in which θ is made equal to $\hat{\theta}(t)$, operational constraints (2.14) and the stability constraint (2.15). Additionally, at each sampling period, parameters estimate and distributed observer dynamics are computed in parallel:

$$\frac{\partial x_a}{\partial t} = -\mathcal{L}_x + s_0(x) + S(x)\hat{\theta} - \mathcal{K}(x-x_a) \tag{2.24}$$

$$\dot{\hat{\theta}} = \text{Proj}\left\{\Gamma\int_0^1 S^T(x)(x-x_a)\,dz,(\overline{\theta},\underline{\theta},\epsilon)\right\}, \tag{2.25}$$

where x is the system state and x_a is the observer state and $\overline{\theta}+\epsilon\geq\hat{\theta}\geq\underline{\theta}-\epsilon$.

2.7 Application to a Fixed-Bed Tubular Bioreactor with Contois Kinetics

Consider now the application of the above techniques to the specific case of a fixed bed tubular bioreactor with two reactions, where the specific growth depends on both substrate and biomass concentrations given by a Contois kinetics model as described in [16, 30]:

$$\frac{\partial x_b}{\partial t} = \mu x_b - k_d x_b \tag{2.26}$$

$$\frac{\partial s}{\partial t} + \frac{u}{L}\frac{\partial s}{\partial z} = -k_1\mu x_b \tag{2.27}$$

$$\frac{\partial x_d}{\partial t} + \frac{u}{L}\frac{\partial x_d}{\partial t} = k_d x_b \tag{2.28}$$

$$\mu = \frac{\bar{\mu}s}{k_c x_b + s}, \tag{2.29}$$

where $x_b(z,t)$ is the biomass concentration, $s(z,t)$ and x_d are the substrate and non-active biomass concentration flowing at velocity $u(t)$, k_1 is the yield coefficient, k_d is the consumption rate. The coefficients associated to the kinetics, $\bar{\mu}$ and k_c are assumed known. Nominal parameter values are given in table 2.1, consumption and yield rates are assumed uncertain $\theta = [k_d \;\; k_1]^T$.

The NMPC algorithm proposed uses a space semi-discrete model obtained by the OCM in $N = 6$ space collocation points. The velocity u is saturated to the values included in the interval $]0.01, 0.5[$ and the state is considered available at the collocation points. It can be shown that using OC with $N = 6$ corresponds to more than 500 finite difference for

Table 2.1: Bioreactor parameters.

Parameters	Value	Units
L	1	m
k_1	0.4	-
k_c	0.4	-
k_d	0.05	h^{-1}
$\bar{\mu}$	0.35	h^{-1}

Table 2.2: Tuned RHC parameters.

Parameters	Value
$\mathcal{T}$	6 h
N_u	6
ρ	250
α	0.001

semi-discretized space in the numerical PDE model solution. Several simulations were made in order to evaluate the importance of, not only the penalty weight of the input in the optimization problem, but also the control horizon, the computational effort (in terms of time spent on simulation) and most importantly the incorporation of the stability constraint in the optimization.

2.7.1 Simulation run 1: Constrained/Unconstrained controller

The series of simulations, as stated before intends to demonstrate the controller performance with different configurations. For the first simulation, figures 2.2-2.4 show respectively the velocity (manipulated variable) $u(t)$, the output $y(t) = s(1,t)$ and the parameters estimates for the *tuned* adaptive NMPC controller of table 2.2 (where ρ is the control effort or penalty weight). Take notice that the parameter estimates converge in the first $30h$ thereby, maintaining the controller performance. The response to a sudden change in set-point has a settling time smaller than $12.5h$ and an approximately $15-20\%$ overshoot. Figure 2.4 shows the estimated parameters, for initial values $\hat{\theta}_1(0) = 0.1, \hat{\theta}_2(0) = 0.3$. The estimates converge to the nominal values given in table 2.1. The estimation gains and the convergence coefficients are given in table 2.3. Figure 2.7 shows the Lyapunov function decreasing for each step set-point change. Because the control law is well tuned the stability condition is not active during all simulation. Finally, figures 2.5 and 2.6 show, respectively substrate and biomass profiles.

Table 2.3: Estimation and convergence parameters.

Parameters	**Value**
k_{px}	$0.1\ I_{N+1}$
k_{ps}	$0.1\ I_{N+1}$
γ_x	0.1
γ_s	1.0×10^{-3}

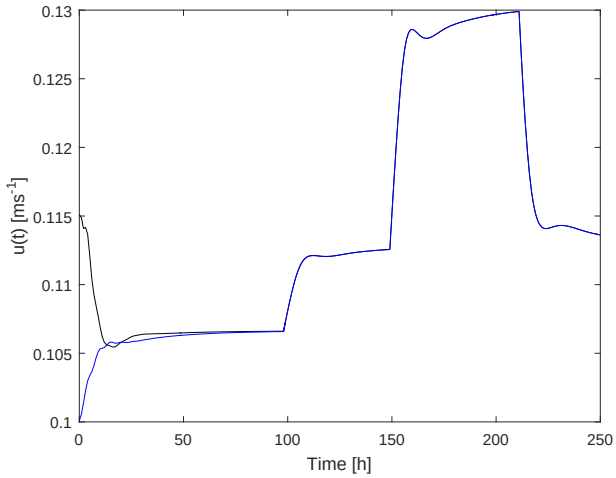

Figure 2.2: Simulation 1: Velocity (input) with stability constraint (black) and without (blue)

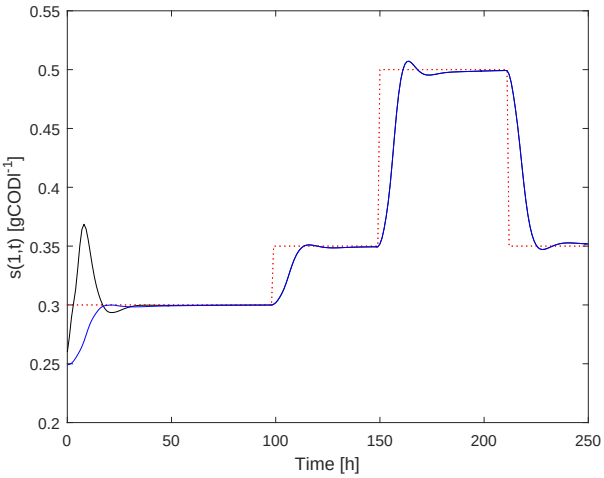

Figure 2.3: Simulation 1: Substrate (output) with stability constraint (black) and without (blue)

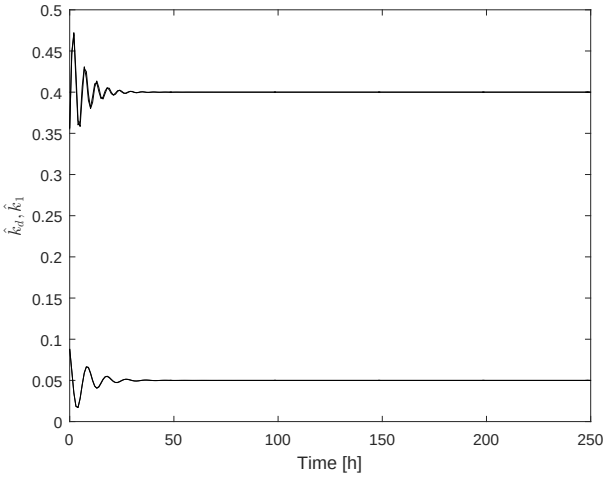

Figure 2.4: Simulation 1: Parameters estimation

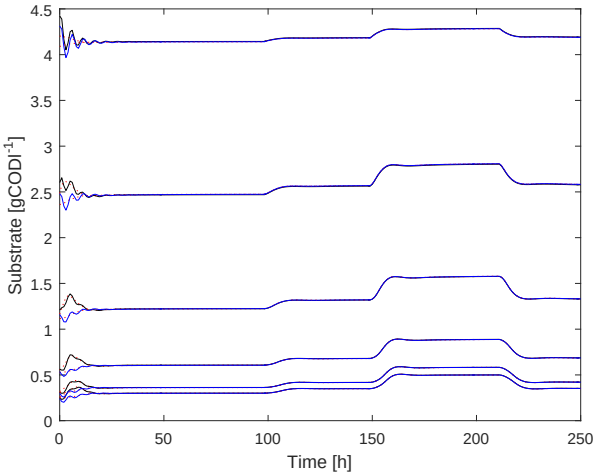

Figure 2.5: Substrate and substrate estimator: $S(t)$, $S_a(t)$. With stability constraint (black) and without (blue)

2.7.2 Simulation run 2: Input weight

For simulation 2, figures 2.8-2.10 show the behavior for a *detuned* control law with a prediction horizon $\mathcal{T} = 1\ h$ and very small input weight $\rho = 1$ (see table 2.4). In this case the stability condition is active during almost every period of time, in order to prevent closed-loop instability. The comparison between the inclusion (or not) of the stability constraint is depicted as previously (black line represents presence of the constraint and blue line represents no stability constraint present). In terms of computational time it has been registered that the inclusion of the stability condition has increased the calculations in 10-25% when compared to the simple unconstrained version. Figure 2.11 shows that the adaptation converges even when closed-loop stability is in risk.

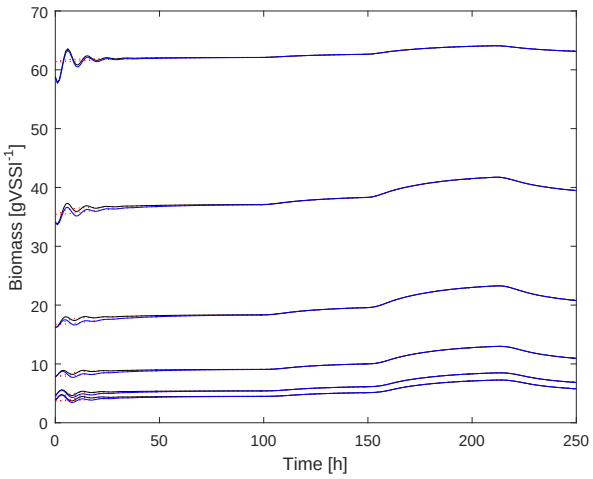

Figure 2.6: Simulation 1: Biomass, $X(t)$, and biomass estimator, $X_a(t)$, with stability constraint (black) and without (blue)

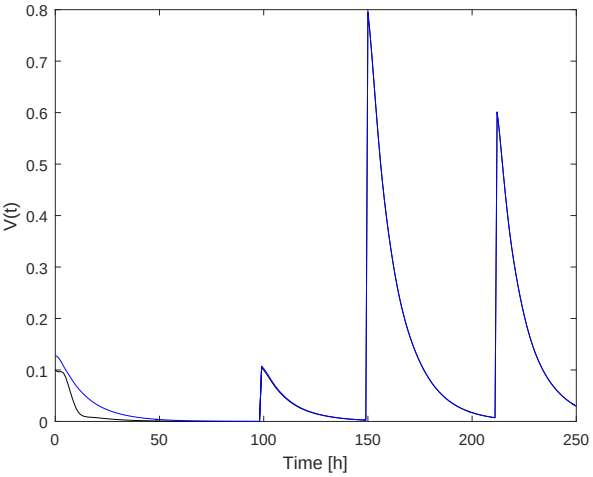

Figure 2.7: Simulation 1: Control Lyapunov Function with stability constraint (black) and without (blue)

Table 2.4: Tuned RHC parameters for simulation 2.

Parameters	**Value**
$\mathcal{T}$	1 h
N_u	6
ρ	1.0
α	0.001

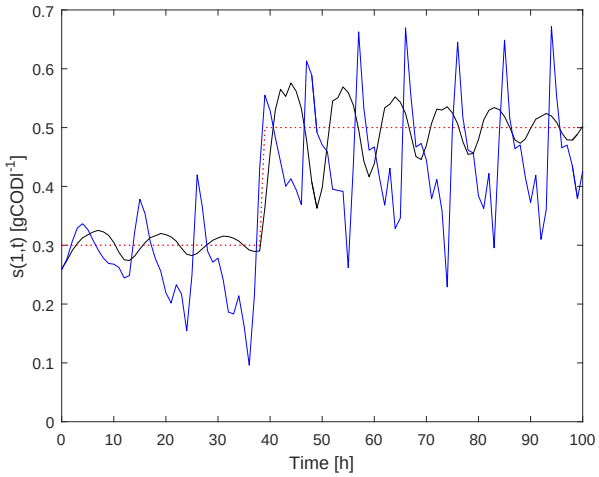

Figure 2.8: Simulation 2: Output, $s(1,t)$, with (black) and without (blue) stability condition.

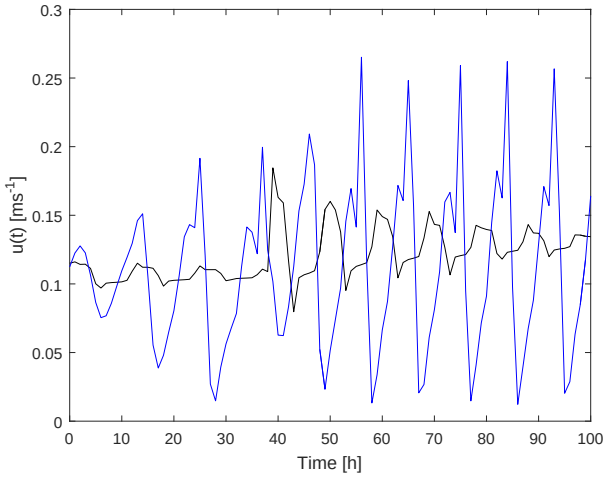

Figure 2.9: Simulation 2: Velocity (input) with (black) and without (blue) stability condition.

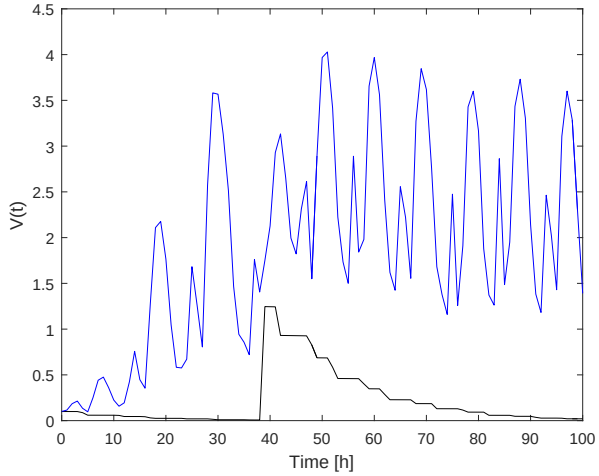

Figure 2.10: Simulation 2: Control Lyapunov function, $V(t)$, decreasing (black) and $V(t)/10$ without stability condition (blue) oscillating.

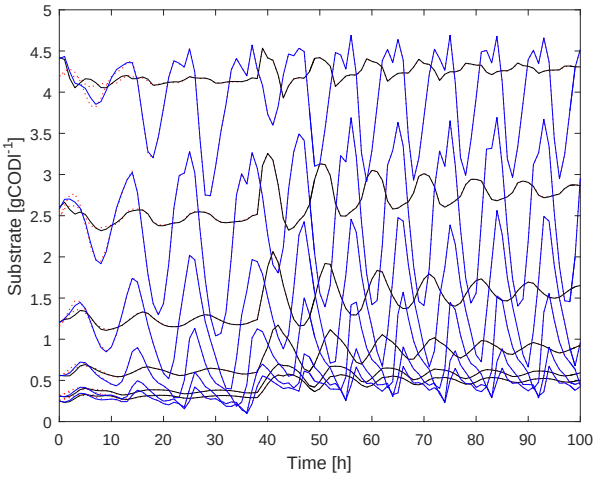

Figure 2.11: Simulation 2: Substrate and Subtract estimation, with (black) and without (blue) stability condition.

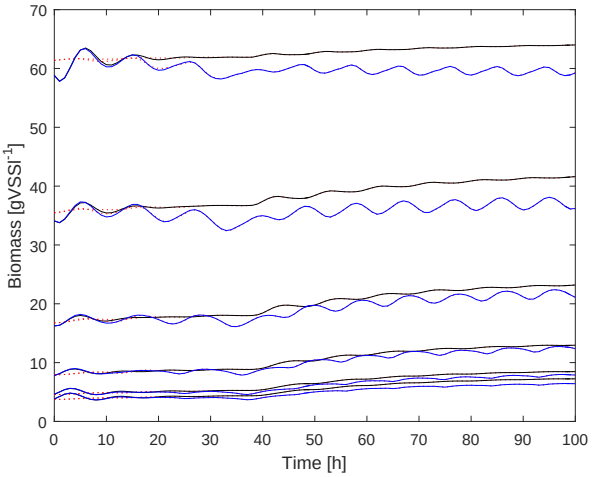

Figure 2.12: Simulation 1: Biomass and Biomass estimation, with (black) and without (blue) stability condition.

2.7.3 Simulation run 3: Additive disturbance

On simulation 3, an additive disturbance (an increase on the inlet substrate concentration S_{in} of 12.5% at 175h) was tested for the constrained version. The results confirm that the inclusion of the stability condition is activated briefly on set-point changes, but also as the disturbance is included, but without degrading the control action. Figures 2.13- 2.15 depict the input, output and control Lyapunov function for this trial run.

2.7.4 Simulation run 4: Prediction horizon

In simulation 4, intends to demonstrate the influence of prediction horizon in the controller behaviour. Figures 2.16- 2.18 depict input, output and CLF for different prediction horizons.

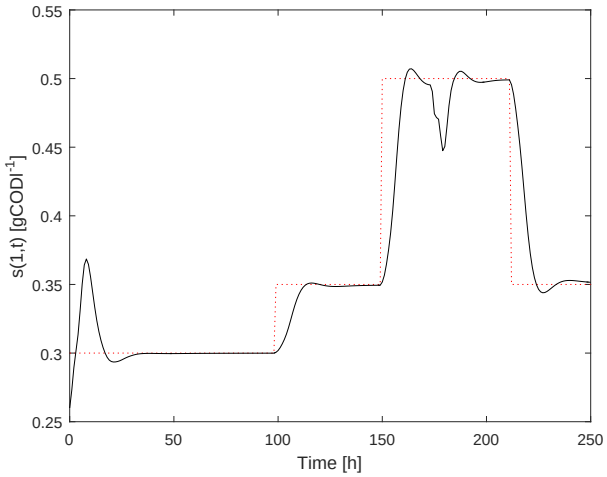

Figure 2.13: Simulation 3: Output, $s(1,t)$.

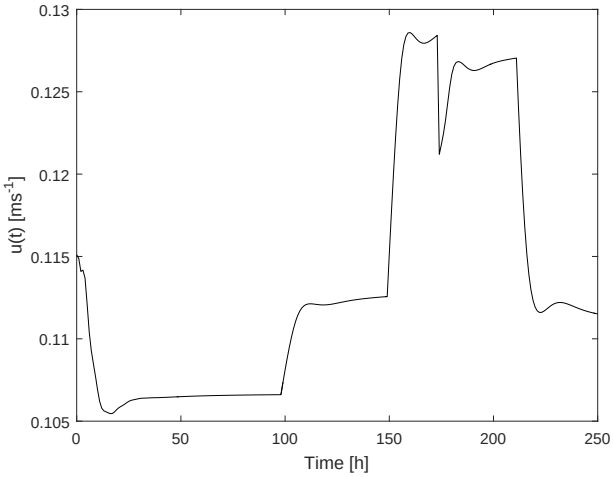

Figure 2.14: Simulation 3: Velocity (input)

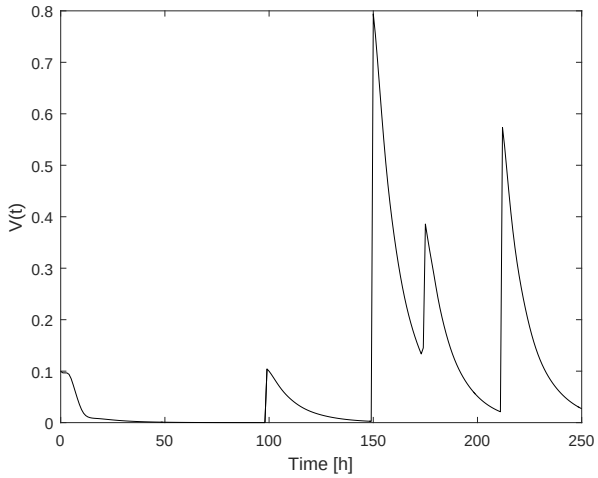

Figure 2.15: Simulation 3: Lyapunov function: $V(t)$ decreasing, $V(t)/10$ without stability condition oscillating.

Table 2.5: Prediction horizon parameter for simulation 4.

Prediction horizon	**Simulation real-time [s]**
10	225
8	208
4	195
2	150

It is shown that smaller horizons tend to give worst performances, as expected, but it also shown that despite poorer performance the controller maintains integrity. In terms of computational time, as could be expected higher horizons tend to demand additional time in simulations, although this increase has been quite moderate (30-50%) as can be seen in table 2.5.

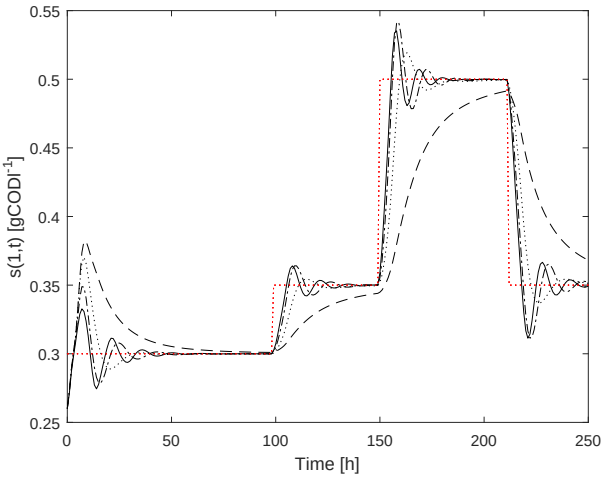

Figure 2.16: Output: $s(1,t)$. Solid line represents higher prediction horizon and dashed line represents lower prediction horizon. Dash-dot and double dotted lines represent intermediate horizons.

2.8 Conclusions

A stable adaptive NMPC formulation for an uncertain class of distributed parameter biosystems was introduced and exemplified in the form of a fixed-bed tubular bioreactor with Contois kinetics with the application of a stabilizing NMPC and LAL. A general stability condition for the combined NMPC nominal formulation with a stable parameter observer was also successfully developed. Low computational effort for closed-loop stability and on-line parameter estimation as a way to deal with uncertainty has been shown to justify this type of approach. A series of simulations were performed in order to

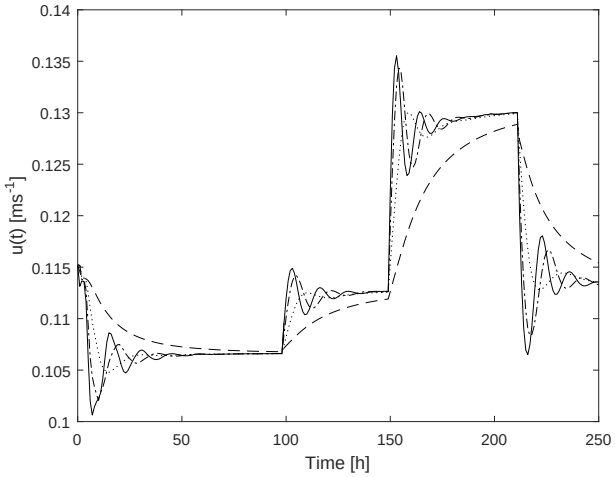

Figure 2.17: Velocity (input). Solid line represents higher prediction horizon and dashed line represents lower prediction horizon. Dash-dot and double dotted lines represent intermediate horizons.

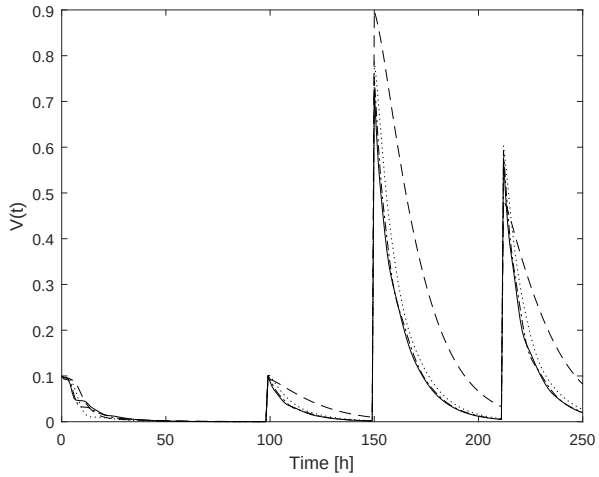

Figure 2.18: Lyapunov function: $V(t)$ Solid line represents higher prediction horizon and dashed line represents lower prediction horizon. Dash-dot and double dotted lines represent intermediate horizons.

demonstrate not only the controller performance, but also the importance of adaptation mechanisms still scarcely applied in nonlinear hyperbolic DPS.

CHAPTER 3

ROBUST POINTWISE MIN-NORM CONTROL

3.1 Introduction

From the previous chapter, PWMN control assumes great relevance as an extension of RCLF. In this chapter the objective is to describe a RPWMN control general result for a class of distributed parameters systems with fluid flow, heat transfer and (bio)reactions processes modelled by a set of hyperbolic and/or parabolic PDE equations including convective and dispersive phenomena. This broad class of models, as described in the previous chapter, arises from physical conservation principles by means of balances of mass, energy and linear momentum and can describe the temperature and (bio)chemical species distribution on moving fluids through pipes (please refer to [20, 83]) and also fluids hydraulics in open pool channels at atmospheric pressure, predicting velocity and mass, space and time behaviour, as explained in [28].

Pointwise Control for nonlinear finite dimensional systems can be found two extremely important text books on nonlinear control such as [40, 111], but also in [95, 96]. For the control of infinite dimensional systems the author in [24] uses semigroups theory (see chapter 1 for further understanding) in linear control for this class of systems and in [27], as seen before, the authors develop robust control methodologies in order to deal with hyperbolic and parabolic distributed systems as the one presented in this chapter.

Different approaches to the use of a PWMN stability condition for predictive control were studied in [95] where a unified framework uses both PWMN and RHC used as a tool for an optimal control scheme (a variation of the Sontag's formula [109]) and [46] where the authors propose, firstly a generalized version of PWMN control that differs on the common version, because it adds an extra term - a guide function - to improve performance in this type of control while the CLF only has to deal with keeping closed-loop stability. They have also introduced the enhanced version for the Generalized PWMN

for NMPC in order to achieve the optimal performance by the selection of the uncertain parameters. In [55] similar CLF techniques for adaptive control of hyperbolic systems are described, namely for a tubular heat exchanger, a distributed collector solar field and a tubular reactor.

In terms of water distribution in open canals control, several studies have been developed and bibliography is now widely available. For instance, in [74] the authors propose to use a combination of robustness with different controller structures and try to achieve a compromise between water resources management and disturbance rejection. In [43, 49, 70] predictive control schemes with adaptation are considered for a multivariate NMPC application to a water canal pool, as the one here presented.

In this chapter, and as stated in chapter 2, the objective is to establish a numerical PWMN control scheme that can be combined in a more general NMPC design. The main goal is to ensure closed-loop stability using moderate values of the prediction horizon (take note that when the horizon tends to zero the PWMN control will be the limit stabilizing solution for NMPC). Simultaneously, the idea is to keep computational effort and off-line computation as low as possible, improving controller performance.

Recall that, as stated in the previous chapter, this type of controller corresponds to a NMPC limit stabilizing solution when the horizon value goes to zero and that can be designed based to the optimization statement:

if $V(\tilde{x}) > l_0$ *then*:

$$\begin{aligned} &\min_{u \in \mathbb{U}} \quad u^T u \\ &s.\,t.\ (2.1)\ and\ \max_{\theta \in \Theta} \{\dot{V}(\tilde{x};\theta)\} + \alpha V(\tilde{x}) < 0, \end{aligned} \tag{3.1}$$

where $V(\tilde{x}) : X \to \mathbb{R}_{\geq 0}$ is, as seen before, a continuously differentiable, positive definite and radially unbounded function in respect to the L_2 norm of $\tilde{x}$.

3.2 Finite Escape Traveling Time Distributed System

The systems here considered present finite escape travelling time, which means that for a given trajectory $\overrightarrow{x}(t)$, the system goes to infinity at a certain time in the future if

$$\lim_{t \to t_{escape}} \| \overrightarrow{x}(t) \| = \infty$$

for some $t_{escape} \in [t_0, \infty[$ [61]. Figure 3.1 depicts a finite escape travelling time system.

Considering the class of PDE models defined in Chapter 2:

$$\begin{aligned} \frac{\partial x(z,t)}{\partial t} + \mathcal{L}(x(z,t);u(t);\theta) &= s(x(z,t);u(t);\theta) \\ \mathcal{M}(x(z,t);u;\theta) &= 0 \end{aligned} \tag{3.2}$$

$$y_k(t) = \int_0^1 b_k(z)\ h_k(x(z,t))dz \qquad (k = 1, ..., p). \tag{3.3}$$

It is important to remember that the $\mathcal{L}$ operator must include convection terms and that it will depend on the manipulated variable $u(t)$. Remember also that if the manipulated variable is space weighted additive in the production term:

$$s(x,u;\theta) = s_x(x;\theta) + s_x(x;\theta)\ w(z)\ u. \tag{3.4}$$

The manipulated variable is, therefore explicit on the model and implicit if appears only in the boundary condition.

As pointed out, these configurations allow the study of tubular reactors and bioreactors [30, 68], but also heat exchangers, solar fields [19], and fluid flow in water distribution canals. The complex dynamics on this class of distributed nonlinear systems, usually strongly dependent with respect to space, may present unstable dynamics as shown in [21], unstable with traveling finite escape time, non minimum-phase behavior shown in [92], *hot spots* characteristics (please read [68]) and fluid flow traveling waves and oscillation as seen in [83].

The optimization problem can be defined as in (2.13), with extended operational constraints defined as:

$$\mathcal{C}(\dot{\eta}(t), \dot{y}(t), \dot{u}(t), \eta(t), y(t), u(t), t) \leq 0 \tag{3.5}$$

and the same stability constraint (2.15).

Given these assumptions, a possibly uncertain distributed hyperbolic system with finite traveling escape time, is given by:

$$\frac{\partial x}{\partial t} + \frac{u(t)}{L}\ \frac{\partial x}{\partial z} = \theta x^2, \tag{3.6}$$

with parameter $\overline{\theta} > \theta > \underline{\theta} > 0$, solving for $x(0,t) = 0$ it yields:

$$x(z,t) = \frac{1}{\varphi(z,0) - \theta t}, \tag{3.7}$$

where $\varphi(z,0) = x^{-1}(z,0)$. Clearly for constant velocity $u > \theta L$ to stabilize the system around $x(z,t) = 0$ if $x(z,0) = 1$, as depicted in figure 3.1.

If $u(t)$ is not constant, an average finite time escape velocity, $\bar{u}_e$, exists and is given by:

$$t_e = \frac{1}{\theta\phi(0)} = \frac{L}{\bar{u}_e}. \tag{3.8}$$

The space stationary profile, defined by $\frac{\partial x}{\partial t} \equiv 0$, can be obtained from:

$$\frac{dx(z)}{dz} = \gamma x^2 \tag{3.9}$$

$$\gamma = \frac{\theta L}{u}. \tag{3.10}$$

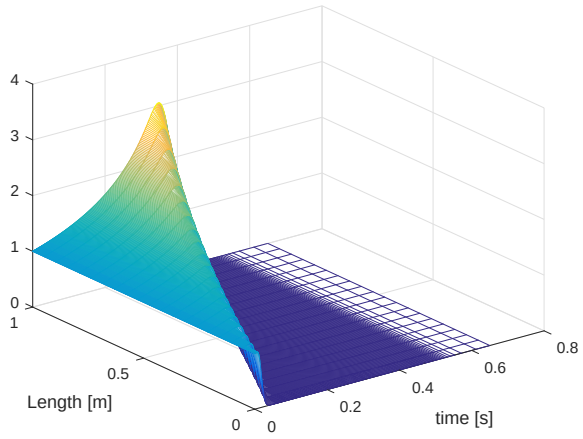

Figure 3.1: Numerical solution, 200 finite differences, for u=2.5 m/s, $\theta = 2$ and $x(z,0) = 1$

For the boundary condition $x(0,t) = x_{in}$, the solution for the stationary state profile in 3.9 is:

$$x_r(z) = \frac{x_{in}}{1 - \gamma_r x_{in} z}. \tag{3.11}$$

Using (3.11) with output set-point defined as $x_r(1,t) = r$:

$$r = \frac{x_{in}}{1 - \gamma_r x_{in}} \quad \text{and} \quad u_r = \frac{\theta\, r x_{in}}{r - x_{in}}, \tag{3.12}$$

with $u_r > 0$ which implies $r > x_{in}$. Substituting (3.12) in (3.11) it yields:

$$x_r = \frac{r x_{in}}{r - rz + x_{in} z} \tag{3.13}$$

for $x_r(0) = x_{in}$ and $x_r(1) = r$.

Stabilizing around the stationary profile, using the CLF:

$$V = \frac{1}{2}\int_0^1 e^2 dz, \qquad e = x - x_r. \tag{3.14}$$

Differentiating (3.14) with respect to time:

$$\dot{V} = -\frac{u}{L}\int_0^1 e\,\frac{\partial x}{\partial z} dz + \int_0^1 e\,\theta x^2 dz. \tag{3.15}$$

The feasible optimization problem can be written as:

$$\begin{aligned} &\min_{u>0} \quad u^2 \\ &s.\,t. \quad \frac{\partial x}{\partial t} + \frac{u}{L}\frac{\partial x}{\partial z} = \theta x^2 - \frac{u}{L}\int_0^1 e\,\frac{\partial x}{\partial z} dz + \max_{\theta}\left\{\theta \int_0^1 e\, x^2 dz\right\} + \alpha\frac{1}{2}\int_0^1 e^2 dz < 0, \end{aligned} \tag{3.16}$$

yielding the following dynamical bound $\max\{\dot{V}\} + \alpha V < 0$. In this case, an analytical solution can be found for (3.16):

$$u = \frac{\check{\Theta}\int_0^1 e\, x^2 dz + \alpha\, V}{\int_0^1 e\,\frac{\partial x}{\partial z} dz} L, \tag{3.17}$$

where $\check{\Theta}$ switches from $\underline{\theta}$ to $\overline{\theta}$ with $sign(\int_0^1 e\ x^2 dz)$. Note that u is well-defined because $\int_0^1 e\ \frac{\partial x}{\partial z} dz \neq 0$ outside the curve level $V(e) = \phi$ and the space operators converge exponentially to zero as $e(z,t) \to 0$.

By observing that the PDE solution $x(z,t) = (x_{in}^{-1} - \theta z L/u(t))^{-1}$, along characteristic curve $\dot{z} = u(t)/L$ which implies $\frac{\partial x}{\partial t} > 0$ and $e(z,t) \neq 0$ for any $z \neq 0$, outside the invariant region. Take note that both the control law and equation (3.17) are coincidental by feedback linearization with $y \equiv V$ with a characteristic index equal to one outside the invariant region, as seen in [24].

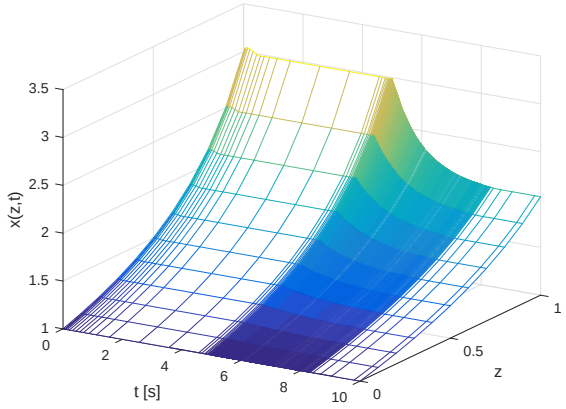

Figure 3.2: State $x(z,t)$ transition, for $\theta = 2$, $\alpha = 2$ and $\phi = 0.0001$.

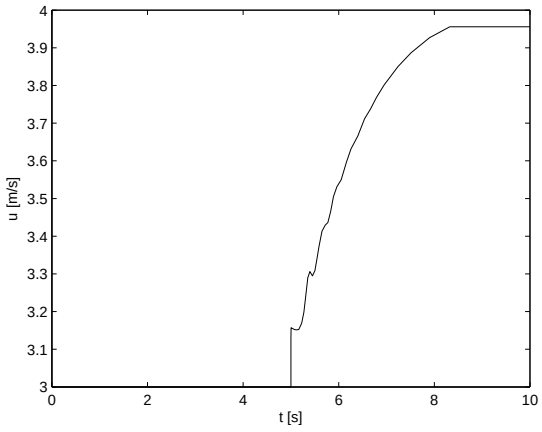

Figure 3.3: Manipulated velocity u [m/s].

Figures 3.2 and 3.3 depicts, respectively state $x(z,t)$ and input $u(t)$ to a sudden setpoint change from at $t = 5\ s$. The simulation results were obtained solving (3.16) by application of the Finite Difference method with 200 space finite differences. Figures 3.4 and 3.5 depict both state and input when the setpoint changes are from 3 to 2 at $t = 5\ s$ and back again to 3 at $t = 10\ s$ for bounded uncertain θ. In both cases u_r is unknown.

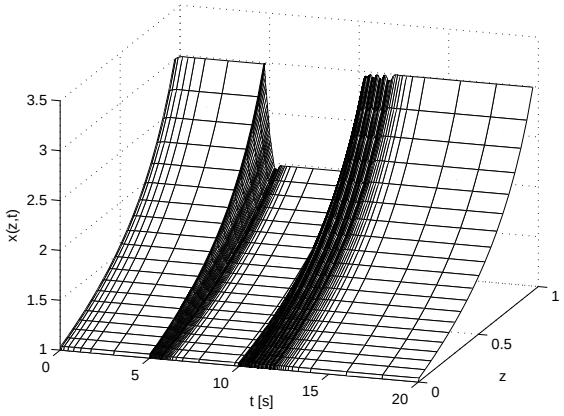

Figure 3.4: Robust state $x(z,t)$ transition, for $\theta \in [1.8\ 2.2]$, $\alpha = 2$ and $\phi = 0.0001$.

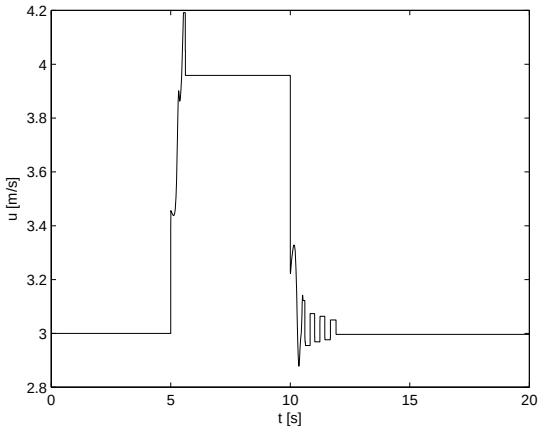

Figure 3.5: Manipulated velocity u [m/s].

3.3 Application to a Water Distribution Canal Pool

One of the best examples of DPS are water distribution canals given the strong nonlinearity and so nonlinear control may be applied. These canals are formed by a sequence of pools separated by gates, as depicted below in figure 3.6. The output variables are the pool level at certain points, the manipulated variables are the gates positions and disturbances are the outlet water flows. This type of system is obviously constrained by the minimum and maximum positions of the gates and also by gate slew rate and the minimum and maximum water level. For this application, the objective is to make the canal pool level to follow a reference in the presence of possible disturbances.

The pool level satisfies the Saint-Venant equations (also known as shallow water equations) which are a set of hyperbolic partial differential equations that embody mass and momentum conservation (derived from Navier-Stokes equations). With that said, considering a single pool model without infiltration, Saint-Venant equations can be given by:

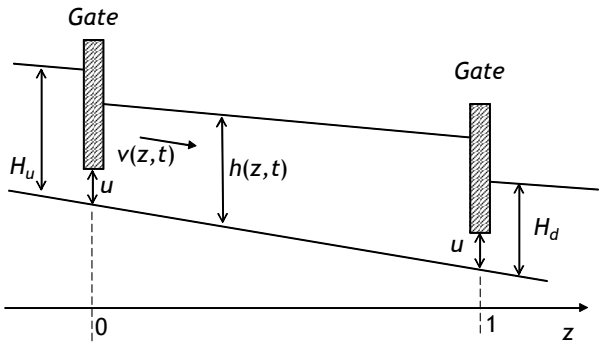

Figure 3.6: Canal pool schematic.

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{v}{L}\frac{\partial h}{\partial z} + \left(\frac{da}{dh}\right)^{-1}\frac{a(h)}{L}\frac{\partial v}{\partial z} &= 0 \\ \frac{\partial v}{\partial t} + \frac{v}{L}\frac{\partial v}{\partial z} + \frac{g}{L}\frac{\partial h}{\partial z} + g(\mathcal{I}(h,v) - \mathcal{J}) &= 0, \end{aligned} \tag{3.18}$$

where $h(z,t)$ and $v(z,t)$ are respectively level and water velocity distributions along space ($z \in [0,1]$) and time. Functions $a(h)$ and friction $\mathcal{I}(h,v)$ are nonlinear functions that represent the wet surface and friction terms, respectively. Finally, the gravity acceleration g, $\mathcal{J}$ is the constant canal slope and L is the pool length. The flow at upstream and downstream gates, that gives the boundary conditions, are:

$$v(0,t) = c_d A_d(u)\sqrt{(2g(H_u(t) - h(0,t)))}/a(h(0,t)) \tag{3.19}$$

$$v(1,t) = c_d A_d(u)\sqrt{(2g(h(1,t) - H_d(t)))}/a(h(1,t)). \tag{3.20}$$

Assume a single trapezoidal reach with two pools, two moving gates positioned in the upstream ends of each pool and a fixed gate at the downstream end. The water elevation immediately before the reach $H_u(t)$ and immediately after the reach $H_d(t)$ are assumed to be known. The physical parameters for the given system are table 3.1.

The control problem is to control water elevation in a pool, using the upstream gate position as control signal, and by doing this solve the PWMN optimization statement in (3.17). The error signal is calculated by doing the difference between the water level measure and the corresponding reference, at some fixed distance from the gate. In order to keep computational burden low, velocity distribution was not incorporated in the optimization process.

3.3.1 Centralized Pointwise Min-Norm Control

Like in the previous chapter, a series of simulation run trials have been made in order to test not only the performance of the controller, to see if stability constraints present

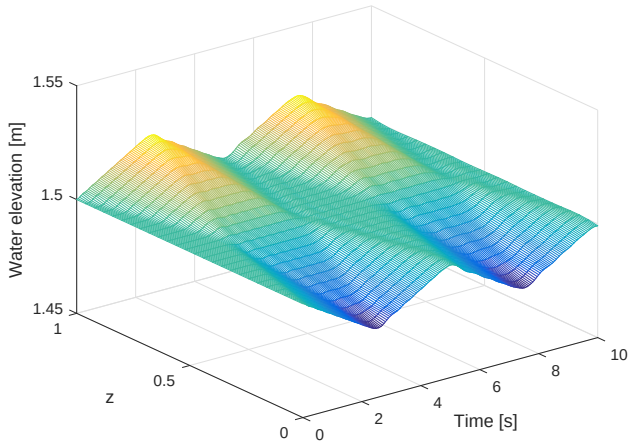

Figure 3.7: Frictionless water oscillation in a closed single pool.

Table 3.1: Pool physical parameters.

Parameter		**Value**	**Units**
Gravitational constant	g	9.8	ms^{-1}
Manning coefficient	n	1.0	$m^{-1}s^{-3}$
Discharge coefficient	c_d	0.6	–
Discharge area	$A_d(u)$	0.49 u	m^2
Bottom width	b	0.15	m
Trapezoid slope	d	0.15	–
Canal slope	$\mathcal{J}$	2×10^{-3}	–
Upstream elevation	H_u	2.0	m
Downstream elevation	H_d	1.0	m

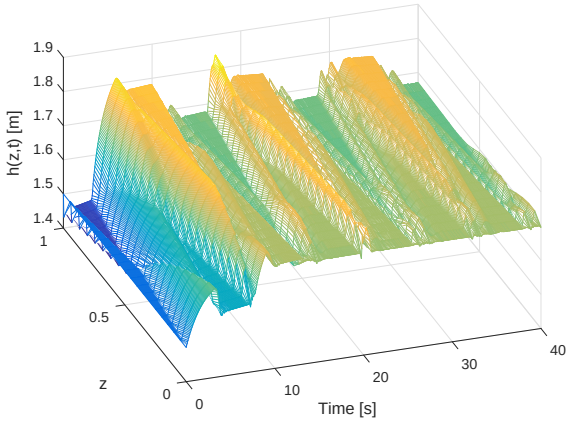

Figure 3.8: Water elevation $h(z,t)$ $[m]$ with real Manning coefficient of 0.018 (concrete canal).

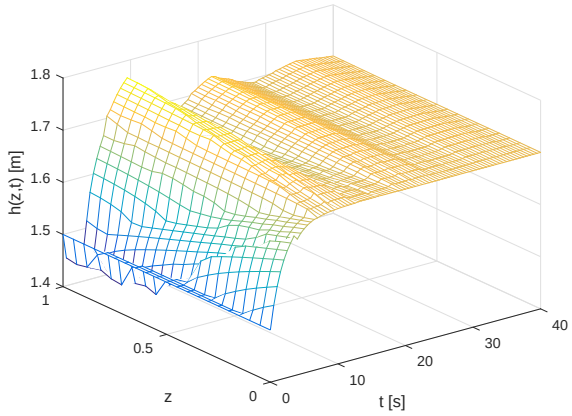

Figure 3.9: Water elevation $h(z,t)$ $[m]$ for a Manning coefficient of 0.1

robustness to different input and/or disturbances and also to evaluate the computational effort.

3.3.1.1 Simulations 1 and 2: additive disturbances in centralized PWMNC scheme for one pool canal

Simulation 1 intends to demonstrate a simple centralized control scheme for RPWMN level control at 1.7 m, upstream, when downstream gate opens from 0 to 0.1 m at $t = 0$ s. The pool was initially at rest with 1.5 m at $z = 0.5$. All numerical results were obtained, as initially stated, with 200 space finite differences. The model space reduction is more extensively detailed in [49]. The chosen controller parameter values are $\alpha = 0.5$ and $\phi = 1 \times 10^{-5}$. Figures 3.10 and 3.12 show, respectively, water elevation and gate manoeuver and figure 3.11 shows water velocity. Make note that the wave back propagation effect on gate opening is very small, causing only a small overshoot.

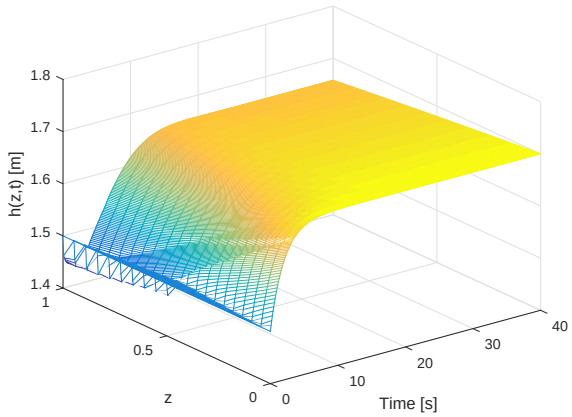

Figure 3.10: Centralized RPWMN scheme: Water elevation $h(z,t)$ $[m]$ for simulation run 1. Manning coefficient of 1.

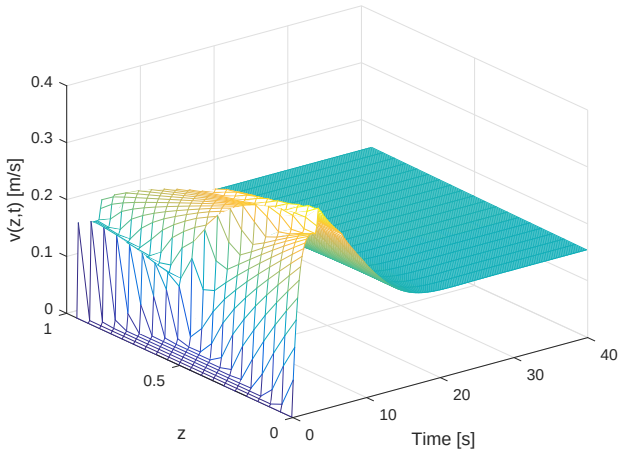

Figure 3.11: Centralized RPWMN scheme: Water velocity $v(z,t)$ [m/s] for simulation 1.

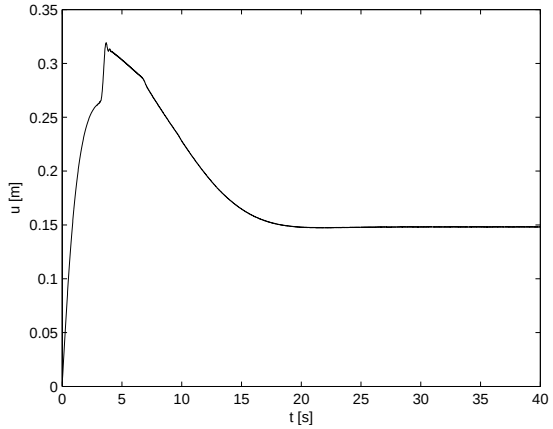

Figure 3.12: Centralized RPWMN scheme: Upstream gate opening u [m] for simulation 1.

As for simulation 2, depicted in figures 3.13 and 3.14 show the same experiment when downstream gate has a delayed opening from 0 to 0.1 m at $t = 10$ s. In this second simulation, contrary to what can be seen in figure 3.10, one can state that there are two combined observable facts: until the downstream gate opens at $t = 10$ s the pool level is increasing, since the downstream gate is closed. This creates a large water rise near it. After $t = 10$ s the downstream gate opens promoting the water drainage. During this period the actuation on the upstream gate opening must act accordingly (figure 3.14) which causes the constraint $u \geq 0$ to be active for a considerable period of time, illustrating how this type of control can handle hard constraints.

3.3.2 Fully Decentralized Pointwise Min-Norm Control

It seems important to test the extension of the SISO case, in the centralized version with one water canal pool and two gates (upstream and downstream), to the MIMO version

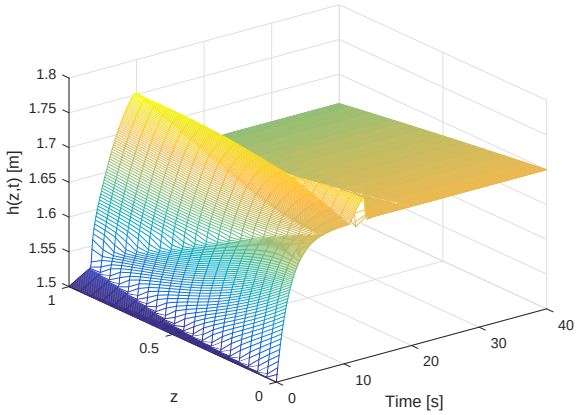

Figure 3.13: Centralized RPWMN scheme: Water elevation $h(z,t)$ $[m]$ for simulation 2.

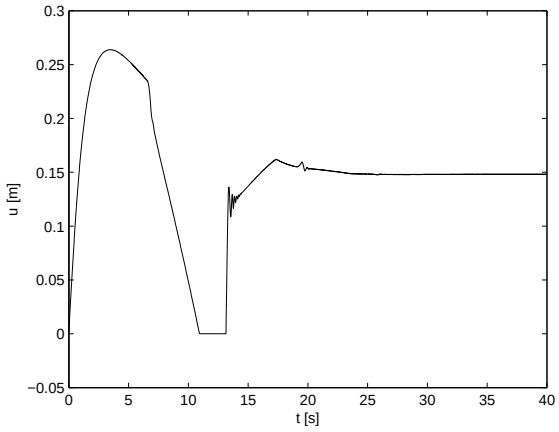

Figure 3.14: Centralized RPWMN scheme: Upstream gate opening u [m] for simulation 2.

of two water canal pools in series with three gates (where the middle gate separates the two canals) as depicted in figure 3.15. The minimization of both local cost functionals is made in sequence at the beginning, moving towards a simultaneous minimization as interactions between both levels increases. The parameter values of the water canal pools were kept as in the centralized trials and are described in table 3.1.

3.3.2.1 Simulations 1, 2 and 3: additive disturbances in decentralized PWMNC scheme for two pool canals

In the decentralized version communication between controllers is non-existing. Each controller deals exclusively with their local cost function and solves the optimization problem independently from a possible optimal global. With that said, figures 3.16- 3.19 show respectively water elevations and upstream and middle gates manoeuver for RPWMN level control at 1.7 m and 1.5 m for canal 1 and 2, as depicted in figure 3.15. Downstream gate is opened at 0.1 m and both pools were initially at rest with 1.5 m at $z = 0.5$, exploring

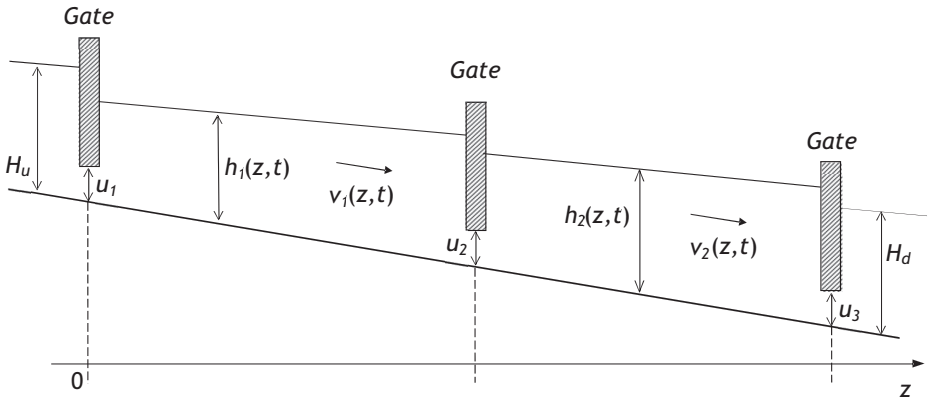

Figure 3.15: Two canal pools schematic.

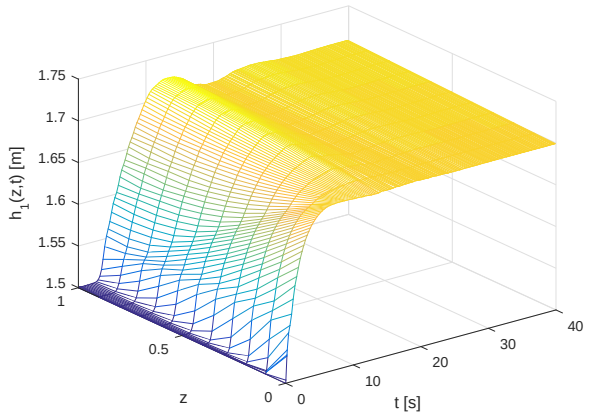

Figure 3.16: Decentralized RPWMN scheme: Water elevation $h_1(z,t)$ $[m]$ (first pool) for simulation 1.

the same conditions of the centralized control scheme for one pool depicted in simulation 1. Controller parameter values were maintained, likewise, at $\alpha = 0.5$ and $\phi = 1 \times 10^{-5}$.

In simulation 2, the main goal was to show both water elevations and upstream and middle gates manoeuver for references of 1.6 and 1.4 m at $t \geq 20$ s in pools 1 and 2, respectively. Figures 3.20- 3.23 shown that both controllers

In simulation 3, the disturbance is made in the downstream gate from 0.1 to 0.12 m. This type of disturbance affects the global installation. For references of 1.6 and 1.4 m in pool 1 and 2, respectively, figures 3.24- 3.27 show that the reference is maintained near the upstream and middle gates, but deviated from the reference a the end of each canal pool as a result of both canal slope and increased velocity imposed by the opening in the downstream gate.

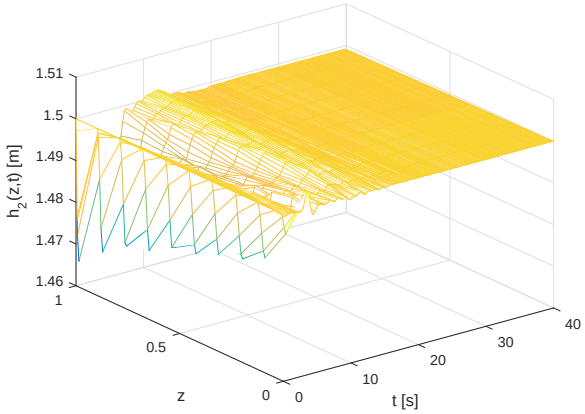

Figure 3.17: Decentralized RPWMN scheme: Water elevation $h_2(z,t)$ $[m]$ (second pool) for simulation 1.

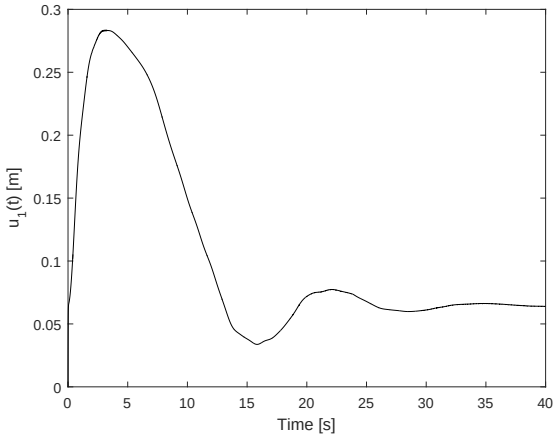

Figure 3.18: Decentralized RPWMN scheme: Upstream gate opening u_1 [m] for simulation 1.

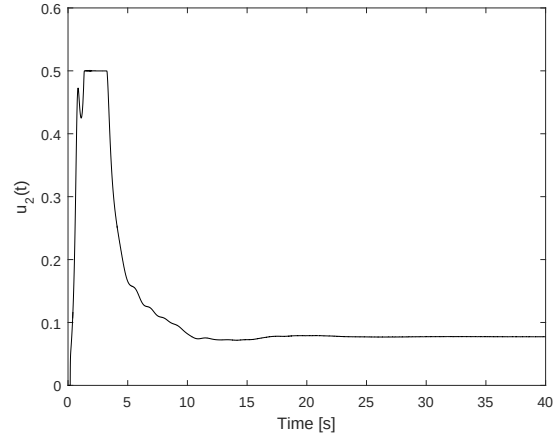

Figure 3.19: Decentralized RPWMN scheme: Middle gate opening u_2 [m] for simulation 1.

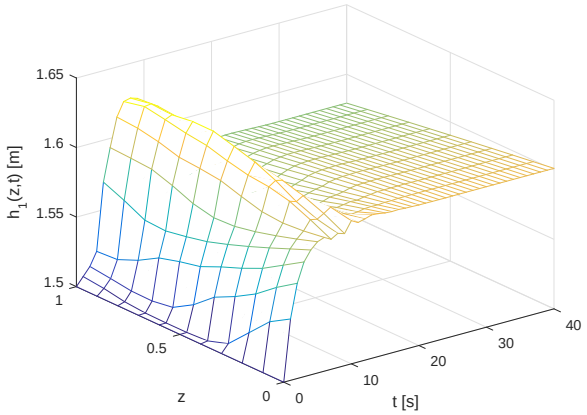

Figure 3.20: Decentralized RPWMN scheme: Water elevation $h_1(z,t)$ $[m]$ (first pool) for simulation 2.

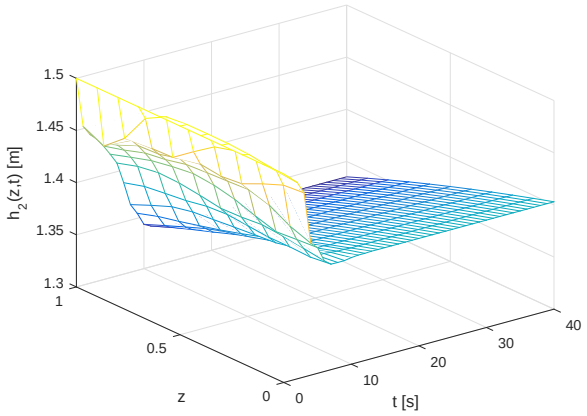

Figure 3.21: Decentralized RPWMN scheme: Water elevation $h_2(z,t)$ $[m]$ (second pool) for simulation 2.

3.4 Conclusions

Using robust pointwise min-norm methodologies it was possible to obtain a general result for distributed fluid flow systems stabilization around a stationary space profile. This obtained optimization statement can be interpreted as a stabilizing limit solution, when included in a predictive receding horizon control formulation, resulting in closed-loop stability for any given horizon. This alternative technique proved to be not only viable, but also with small increase on the computational effort. An application on water distribution canal pool RPWMN control form a basis of more complex studies on canal engineered architectures that by combining robust and predictive design methods can achieve a fair compromise between water resources management and disturbance rejection. Both SISO and MIMO cases were studied in centralized and decentralized control architectures, respectively. In the MIMO case it was observed that controllers appeared to be implemented in sequence, but evolving to simultaneous optimization as the interactions tended to

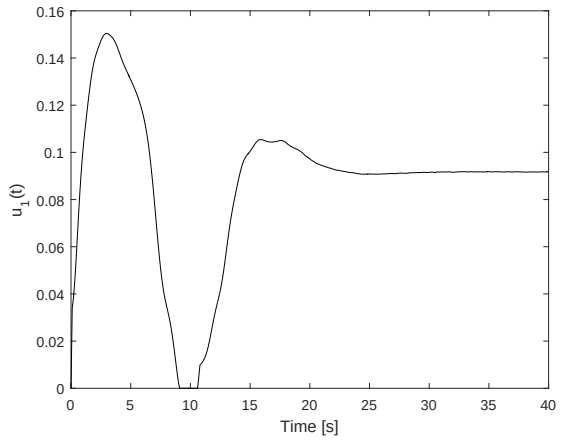

Figure 3.22: Decentralized RPWMN scheme: Upstream gate opening u_1 [m] for simulation 2.

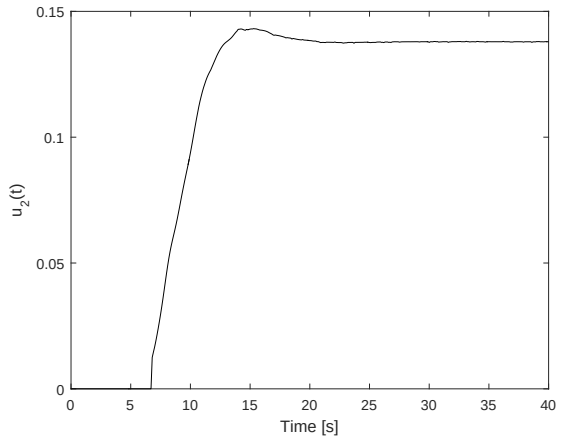

Figure 3.23: Decentralized RPWMN scheme: Middle gate opening u_2 [m] for simulation 2.

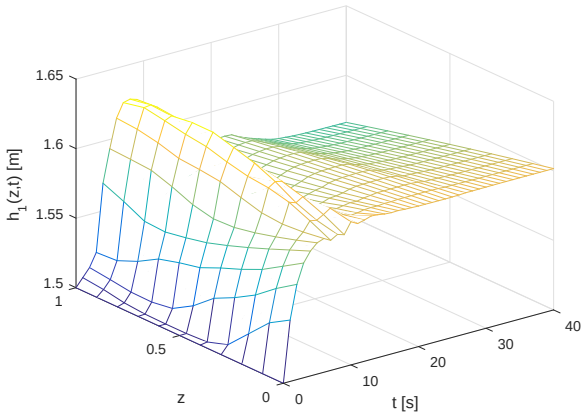

Figure 3.24: Decentralized RPWMN scheme: Water elevation $h_1(z,t)$ $[m]$ (first pool) for simulation 3.

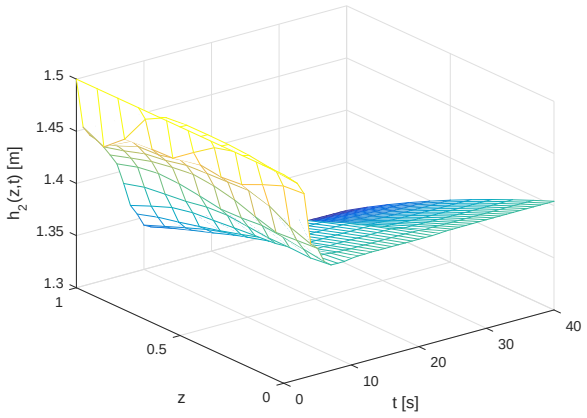

Figure 3.25: Decentralized RPWMN scheme: Water elevation $h_2(z,t)$ $[m]$ (second pool) for simulation 3.

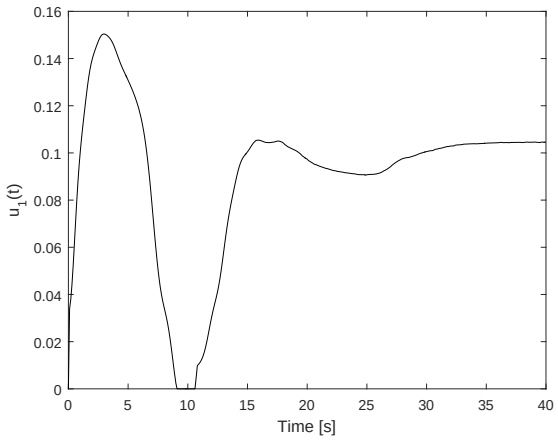

Figure 3.26: Decentralized RPWMN scheme: Upstream gate opening u_1 [m] for simulation 3.

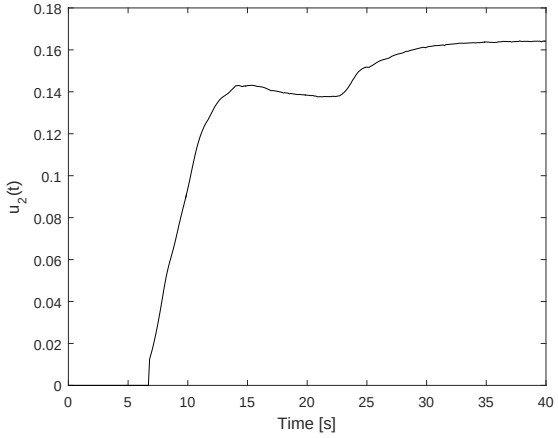

Figure 3.27: Decentralized RPWMN scheme: Middle gate opening u_2 [m] for simulation 3.

increase. The extension to the MIMO case also showed that computational effort seems to be similar behavior to the centralized version.

CHAPTER 4

DISTRIBUTED MODEL PREDICTIVE CONTROL FOR SERIALLY CONNECTED SYSTEMS

4.1 Introduction

In this chapter the idea of cMPC is to be abandoned due to the high complexity of the system under study, making the global optimal solution for cMPC, most of the times, unfeasible. The application of decentralized and distributed MPC schemes have proven to be the best choice for these type of multivariable systems as an application to intelligent infrastructures.

The use of cooperative DMPC as described in chapter 1, section 1.3 is intended in order to obtain global optimal performance, a Pareto Optimal solution. A decentralized scheme is applied to the system with the objective of achieving local performance indexes that are best suited to globally control in closed-loop large scale and geographically expansive systems. A behaviorial induction approach is also introduced and discussed in the model predictive control framework. The algorithm is to be fully developed for generic serially chained systems.

With the cMPC regulation problem defined in equations (4.1)-(4.4), the feasible optimization problem is solved as a way to find the subsystems inputs $u_i(k)$ at each instant in discrete time k as described by the authors in [119]. The discrete global system model is considered to be linear [104] and the cost function $J(k)$ is quadratic with symmetric weighting matrices Q_i and R_i.

$$\min_{U} J(X, U; x(k)) = \sum_{i} w_i J_i(x_i, u_i; x_i(k)) \tag{4.1}$$

$$s.\,t.\; x(k+1) = Ax(k) + Bu(k) \tag{4.2}$$

$$u_i \in \mathbb{U}_i, \tag{4.3}$$

where $w_i > 0$, $\sum_i w_i = 1$ and $\mathbb{U}_i$ is an admissible control convex set.

$$\begin{aligned} J_i(k) &= \sum_{j=1}^{N} \left(x_{i,k+j}^T Q_i x_{i,k+j} + u_{i,k+j-1}^T R_i u_{i,k+j-1} \right) \\ X &= [x^T(k+1) \cdots x^T(k+N)]^T \\ U &= [u^T(k) \cdots u^T(k+N-1)]^T \\ Q_i &\geq 0, \qquad R_i > 0, \end{aligned} \tag{4.4}$$

where $x_{i,k}$ is the subsystem state at k and is known and also that the pair $(A_i, Q_i^{1/2})$ is detectable [104]. Note that $x_{i,k}$ and $u_{i,k}$ denotes the state and input i, at instant k.

Centralized MPC achieves the best attainable performance, Pareto Optimal, due to the fact that the effects of interconnections among subsystems are accounted for exactly. The optimization takes into account the conflicts between control objectives and these are solved optimally. The controller is defined by implementing the first input of the corresponding solution for the optimization problem in a receding horizon control scheme.

In a distributed regulator version of MPC (4.5)-(4.9) the global model is now partitioned in smaller subsystems in order to replace the global cost function by local ones making it easier to solve optimally. These local functions measure the impact of local control agents action and make use of a communication based strategy, based upon information exchange constrains. Each subsystem uses a local cost function that iterates L times before obtaining the control input $u_i^L(k)$. Take note that at each iteration the optimization problem is solved. This means that for each iteration the objective is to find $U_i^l(k)$ for $x_\hbar^{l-1}(k)$ and $u_\hbar^{l-1}(k)$ which are broadcasted by the other subsystem controllers ($\hbar$ indexes the set $\aleph_i$, that includes all subsystems directly connect to subsystem Σ_i). If there exists convergence in this communication based iteration, different types of equilibrium can be achieved by adjusting the weights α_h (h indexes the set $\aleph_i \cup \Sigma_i$). If the solution induces egocentric behavior, $\alpha_{h \neq i} = \epsilon$, with $\epsilon \ll 1$, except for $\alpha_{h=i} \gg \epsilon$ then at limit ($\epsilon \to 0$) a Nash equilibrium is achieved (make note that Nash equilibria has been presented in chapter 1, section 1.3 as the type of solution for the non-cooperative type of controllers) which can result in potentially poor closed-loop performance or even instability. This is due to competitiveness between different controllers instead of a cooperative solution. On the other hand, if altruistic behavior is present, $\alpha_h \neq 0$, then all iterates generated by a cooperation algorithm are strictly feasible which will satisfy all the local constraints and the result is a nominal distributed closed-loop stable control law, as it will be shown for the particular algorithm developed in this chapter. Finally for a pure altruist scenario, α_h will be all equal.

In summary, cooperative distributed control systems can be seen as a set of computational agents that cooperate with their neighbours, in an open information interchange infrastructure, to achieve local performance indexes, suitable for global control optimization.

$$\min_{U_i^l} \bar{J}_i^l(X_h, U_i; x_h(k)) = \min_{U_i^l} \sum_h \alpha_h J_{h,i}(k) \tag{4.5}$$

$$s.\,t.\; x_h(k+1) = A_{hh}x_i(k) + B_{hh}u_i(k) + \sum_{\hbar} \left(A_{h\hbar}\; x_{\hbar}^{l-1}(k) + B_{h\hbar}\; u_{\hbar}^{l-1}(k) \right) \tag{4.6}$$

$$x_i(k+N+1)...x_i(k+N+P) = 0 \tag{4.7}$$

$$u_i \in \mathbb{U}_i, \tag{4.8}$$

where $l = 1,\ ...,\ L$ and

$$J_{h,i}(k) = \sum_{j=1}^{N} \left(x_{h,k+j}^T Q_i x_{h,k+j} \right) + \sum_{j=0}^{N-1} u_{i,k+j}^T R_i u_{i,k+j}. \tag{4.9}$$

Moreover

$$\begin{aligned} X_h &= \left[x_h^T(k+1) \cdots x_h^T(k+N) \right]^T \\ U_i &= \left[u_i^T(k) \cdots u_i^T(k+N-1) \right]^T. \end{aligned} \tag{4.10}$$

The setup includes P zero state terminal constraints that have a crucial role in global closed loop stability as to be explained further ahead. In [22] the authors discuss in detail how general conditions for cooperative iterates convergence and solution uniqueness, by the computational agents, is applied.

4.2 Distributed Stabilizing Input/Output Receding Horizon Control

SIORHC is a receding tracking horizon control algorithm equipped with stabilizing zero terminal constraints for moderate values of the prediction horizon dimension, as seen in figure 4.2. The centralized original algorithm is presented in [88] and early results and extensions to MIMO systems can be found respectively in [89] and [48]. In this section a distributed version of the algorithm is derived for serially chained systems based on neighborhood optimization, as presented in [127], as a particular case of the dMPC discussed above when there are no input constraints, $\mathbb{U}_i \equiv \mathbb{R}^{m_i}$, and when zero equality terminal constraints at the end of the prediction horizon, forcing the outputs to take a particular value, are included. This constraint will play a major role in guaranteeing global closed-loop stability, as shown in the previous section of this chapter. In fact, in this case the assumption of a linear model and unbound inputs yields, by solving the stated optimization problem for each local control agent, an analytical iterative solution that converges to the unique optimal distributed solution, providing that a check condition can be verified *a priori.* This cooperative behavior among the control agents can be induced, as stated before, by adjusting weights on local cost functions yielding different overall performances. The decentralized solution exploits the assumption that in distributed

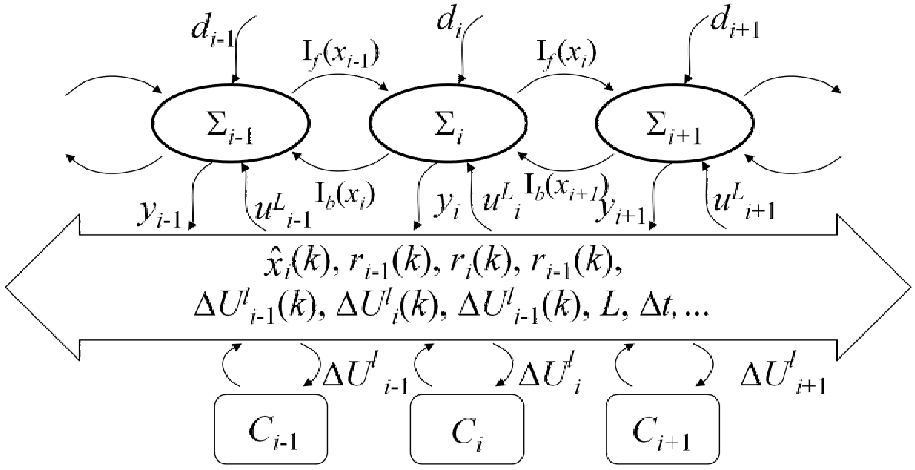

Figure 4.1: Agent shared information service, C_i; Information interchanged by system i, Σ_i, is depicted in the service bus. $I_{f(b)}$ denotes forward (backward) interactions and the d_i denotes disturbances.

systems the interactions between the subsystems can be sparse, which means that the local models dimension $(\aleph_i \cup \Sigma_i)$ can be much smaller than the overall system dimension. This allows that the original highly dimensional problem can be reduced to a set of simpler low dimensional problems to be solved in parallel computation. A generic Service Oriented Architecture, a open computational infrastructure, where the agents can cooperate by subscribing services is depicted in figure 4.1. This figure stresses the shared and broadcasted information during the iterative cooperation by each control agent. Technological issues on cooperative service oriented open process control can be read in [94].

Consider the extended or augmented linear state model:

$$\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \Delta u_k \end{bmatrix}, \tag{4.11}$$

where A, B and C have $(n \times n)$, $(n \times m)$ e $(p \times n)$ dimensions, x_k, y_k and Δu_k are respectively the outputs, the state and moves sequences, *i. e.* the incremental inputs.

For this system the predicted outputs at $k + j$ can be obtained from:

$$\begin{aligned} \hat{y}(k+j) &= \sum_{i=0}^{j-1} C\, A^{j-i-1} B \Delta u(k+i) + \hat{y}_0(k+j) \\ \hat{y}_0(k+j) &= C\, A^j\, x(k) = \Gamma\, x(k), \end{aligned} \tag{4.12}$$

where $\hat{y}_0$ is the output predicted value without moves. For $j = 1...N, N+1, ..., N+P$, (4.12) yields the following predictive equations:

$$\begin{aligned} \hat{Y}_N &= G_N \Delta U + \Gamma_N x(k) = G_N \Delta U + \hat{Y}_{0N} \\ \hat{Y}_P &= G_P \Delta U + \Gamma_P x(k) = G_P \Delta U + \hat{Y}_{0P}, \end{aligned} \tag{4.13}$$

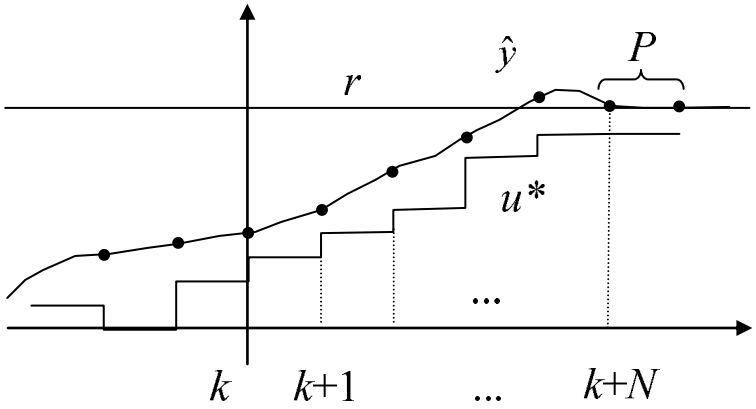

Figure 4.2: SIORHC prediction horizon with contraints.

with:

$$\begin{aligned} \hat{Y}_N &= [y_{k+1} \cdots y_{k+N}]^T \\ \hat{Y}_P &= [y_{k+N+1} \cdots y_{k+N+P}]^T \\ \Delta U &= [\Delta u_k \cdots \Delta u_{k+N-1}]^T. \end{aligned} \tag{4.14}$$

Equation (4.13) can be recast by considering only input related interactions between neighboring connected systems:

$$\begin{aligned} \hat{Y}_{Ni} &= \sum_h G_{Nih} \Delta U_h + \hat{Y}_{0Ni} \\ \hat{Y}_{Pi} &= \sum_h G_{Pih} \Delta U_h + \hat{Y}_{0Pi}. \end{aligned} \tag{4.15}$$

Local controllers are obtained by minimizing the local cost function for the i subsystem:

$$\begin{aligned} \bar{J}_i &= \sum_g \left(\alpha_{i,g} \sum_{j=1}^{N} e_{g,k+j}^T \, Q_{i,g} \, e_{g,k+j} \right) \\ &\quad + \sum_{j=0}^{N-1} \Delta u_{i,k+j}^T \, R_i \, \Delta u_{i,k+j}, \end{aligned} \tag{4.16}$$

with zero terminal horizon constrain given by, as can be seen in figure (4.2):

$$[y_{i,k+N+1} \ \cdots \ y_{i,k+N+P}]^T = [r_{i,k+N+1} \ \cdots \ r_{i,k+N+P}]^T, \tag{4.17}$$

where $e_{i,k} = r_{i,k} - y_{i,k}$, is the subsystem i error, at instant k, in relation with the reference sequence, $r_{i,k}$, and $Q_{i,g} \geq 0$ and $R_i > 0$ are symmetric weighting matrices. Weights $\alpha_{i,g}$ can change agents behavior as explained before. Integer variable g has the same meaning of h.

Equivalently minimizing $\bar{J}_i$ with respect to ΔU_i can be written as:

$$\begin{aligned} \min_{\Delta U_i} \bar{J}_i &= \sum_g \alpha_{i,g} \| Y_{RNg} - \hat{Y}_{Ng} \|^2_{Q_{i,g}} \\ &\quad + \| \Delta U_i \|^2_{R_i} \\ &\quad s.\,t. \ \ \hat{Y}_{Pi} = Y_{RPi}. \end{aligned} \tag{4.18}$$

The stated Quadratic Programming (QP) optimization problem with constrains can now be solved by finding the vector ΔU_i that minimizes the Lagrangian:

$$\mathscr{L}_i := \sum_g \alpha_{i,g} \| Y_{RNg} - \hat{Y}_{Ng} \|^2_{Q_{i,g}} + \| \Delta U_i \|^2_{R_i} + \\ [\hat{Y}_{Pi} - Y_{RPi}]^T \lambda_i, \tag{4.19}$$

using (4.15) then:

$$\begin{aligned} \mathscr{L}_i &= \sum_g \alpha_{i,g} \| E_{Ng} - \sum_h G_{Ngh} \Delta U_h \|^2_{Q_{i,g}} \\ &\quad + \| \Delta U_i \|^2_{R_i} + \left[E_{Pi} + \sum_h G_{Pih} \Delta U_h \right]^T \lambda_i, \end{aligned} \tag{4.20}$$

where:

$$\begin{aligned} E_{Ni} &= Y_{RNi} - \hat{Y}_{0Ni} \\ E_{Pi} &= Y_{RPi} - \hat{Y}_{0Pi} \end{aligned} \tag{4.21}$$

and λ_i is a $P.p_i$ column vector of Lagrange multipliers. The conditions to find the minimum are:

$$\begin{aligned} \frac{\partial L_i}{\partial \Delta U_i} &= 0 \\ \frac{\partial L_i}{\partial \lambda_i} &= 0. \end{aligned} \tag{4.22}$$

Using (4.22) the solution amounts to solve an algebraic set of linear equations $\Phi \Delta U = \Psi(k)$. Where $\Delta U(k)$ is an unknown vector collecting all ΔU_i, Φ is a constant matrix and $\Psi(k)$ is a variable matrix in k.

4.2.1 Serially connected systems solution

Consider the case when the subsystems are connected in series and coupled only by the state variables of its left and right neighbours, as depicted in figure 4.1. Solving (4.22), for all the subsystems, yields the building block matrices for the distributed SIORHC serially connected systems solution, *viz.*:

$$\begin{aligned} \Phi &= [\textit{block pentadiagonal} \\ &\quad \left\{ \cdots \Phi_{i(i-2)} \ \Phi_{i(i-1)} \ \Phi_{ii} \ \Phi_{i(i+1)} \ \Phi_{i(i+2)} \cdots \right\}] \end{aligned} \tag{4.23}$$

$$\Psi = \begin{bmatrix} \vdots \\ \Psi_i \\ \vdots \end{bmatrix}, \tag{4.24}$$

$$\begin{aligned}
\Phi_{i(i-2)} &= S_i \left(\alpha_{i-1} G^T_{N(i-1)i} Q_{i-1} G_{N(i-1)(i-2)} \right) \\
\Phi_{i(i-1)} &= S_i \Big(\alpha_{i-1} G^T_{N(i-1)i} Q_{i-1} G_{N(i-1)(i-1)} \\
&\quad + \alpha_i G^T_{Nii} Q_i G_{Ni(i-1)} \Big) + G^T_{Pii} W_i G_{Pi(i-1)} \\
\Phi_{ii} &= M_i \\
\Phi_{i(i+1)} &= S_i \Big(\alpha_i G^T_{Nii} Q_i G_{Ni(i+1)} \\
&\quad + \alpha_{i+1} G^T_{N(i+1)i} Q_{i+1} G_{N(i+1)(i+1)} \Big) \\
&\quad + G^T_{Pii} W_i G_{Pi(i+1)} \\
\Phi_{i(i+1)} &= S_i \left(G^T_{N(i+1)i} Q_{i+1} G_{N(i+1)(i+2)} \right),
\end{aligned} \tag{4.25}$$

$$\begin{aligned}
\Psi_i &= S_i \sum_{h=-1}^{1} \alpha_{i+h} G^T_{N(i+h)i} Q_{i+h} E_{N(i+h)} \\
&\quad + G^T_{Pii} W_i E_{Pi},
\end{aligned} \tag{4.26}$$

where $h = \{-1, 0, 1\}$ indexes the subsystem immediately at the left of system i, itself (i) and immediately at the right respectively, and:

$$S_i = I - G^T_{Pii} W_i G_{Pii} M_i^{-1} \tag{4.27}$$

$$W_i = \left(G_{Pii} M_i^{-1} G^T_{Pii} \right)^{-1} \tag{4.28}$$

and

$$M_i = \sum_{h=-1}^{1} \alpha_{i+h} G^T_{N(i+h)i} Q_{i+h} G_{N(i+h)i} + R_i. \tag{4.29}$$

Make note that for the solution by the Lagrangian multiplier method to exist $Pp_i < Nm_i$ and the rank of G_{Pii} should be equal to Pp_i. Remark also that if simultaneous calculation were possible then $\Delta U = \Phi^{-1}\Psi$.

4.3 Parallel Iterated Solution

Each distributed SIORHC control agent $C_i(k)$ must iterate during the sampling time period k the following difference equation:

$$\Delta U_i(l) = \Phi_{ii}^{-1} \left(\Psi_i - \sum_{\hbar} \Phi_{ij} \Delta U_j(l-1) \right), \tag{4.30}$$

where $l = 1, 2, ..., L$, indexes the number of iterations during each sampling time period, for simplicity of notation the time index has been omitted in (4.30). Starting with

the initial condition:

$$\Delta U_j(0) = \left[\begin{array}{cccc} \Delta u_j^L(k+1|k-1) & ... & \Delta u_j^L(k+N-1|k-1) & 0 \end{array} \right]^T \tag{4.31}$$

that is the control input sequence obtained in the last time $k-1$ for future instants $k+1, ..., k+N-1$ and because there is no prediction at $k+N|k-1$ the last array position is filled with a zero row block, meaning no moves at the horizon end.

The iterates will converge to the algebraic fully-iterated solution if the following spectrum radius condition hold: $\forall_i \ |\lambda_i| < 1$, where λ_i are the difference equation matrix eigenvalues given by:

$$|\lambda_i I - \Phi_d^{-1}\Phi_{nd}| = 0, \tag{4.32}$$

where:

$$\Phi_d + \Phi_{nd} = \Phi, \tag{4.33}$$

with $\Phi_d = [diag\ \{\cdots\ \Phi_{ii}\ \cdots\}]$.

The spectral radius condition must be checked and it depends on the control action parameters.

4.4 Nominal Closed-Loop Stability

In this section a nominal closed-loop stability for distributed predictive control with zero terminal equality constraints is proved for the fully-iterated algorithm considering that the prediction models $\hat{Y}_{Pi}$ $(\forall\ i)$ contains all the interactions terms between subsystems. The proof is accomplished only for serially connected systems by the sake of clarity.

Consider non-zero initial conditions, $x(0) \neq 0$, and that under closed-loop, the system state is to be steered to zero with the fully-iterated distributed SIORHC controller. To prove close-loop stability consider the cost function for the generic subsystem i:

$$\begin{aligned} \min_{\Delta U_i} \bar{J}_i(k) &= \sum_{h=-1}^{1} \alpha_{i+h} \sum_{j=1}^{N} \|\hat{y}_{i+h}(k+j)\|^2_{\bar{Q}_{(i+h)j}} \\ &\quad + \sum_{j=0}^{N-1} \|\Delta u_i(k+j)\|^2_{\bar{R}_{i(j+1)}}, \end{aligned} \tag{4.34}$$

where:

$$\begin{aligned} \|\hat{y}_i(k+j)\|^2_{\bar{Q}_{ij}} &= \hat{y}_i(k+j)^T Q_{ij}\ \hat{y}_i(k+j) \\ \hat{y}_i(k+j) &= C_i\ \hat{x}_i(k+j). \end{aligned} \tag{4.35}$$

In terms of the state, this can be written as:

$$\begin{aligned} \min_{\Delta U_i} \bar{J}_i(k) &= \sum_{h=-1}^{1} \alpha_{i+h} \sum_{j=1}^{N} \|\hat{x}_{i+h}(k+j)\|^2_{\bar{Q}_{(i+h)j}} \\ &\quad + \sum_{j=0}^{N-1} \|\Delta u_i(k+j)\|^2_{\bar{R}_{i(j+1)}} \end{aligned} \tag{4.36}$$

$$\mathbb{Q}_{ij} = C_i^T Q_{ij} C_i. \tag{4.37}$$

The optimal control sequence $\bar{J}_i^*(k)$ is obtained by replacing the control increments with the optimal ones $\Delta U_i^*(k)$ and it can be written as:

$$\begin{aligned}\bar{J}_i^*(k) = &\sum_{h=-1}^{1} \alpha_{i+h} \|\hat{x}_{i+h}(k+1)\|^2_{\mathbb{Q}_{(i+h)1}} \\ &+ \sum_{h=-1}^{1} \alpha_{i+h} \sum_{j=2}^{N} (\|\hat{x}_{i+h}(k+j)\|^2_{\mathbb{Q}_{(i+h)j}} \\ &+ \|\Delta u_i^*(k)\|^2_{R_{i1}} + \sum_{j=1}^{N-1} \|\Delta u_i^*(k+j)\|^2_{R_{i(j+1)}}.\end{aligned} \tag{4.38}$$

Applying the following feasible control sequence at $k+1$, $\Delta U_i'(k+1) \equiv \{\Delta u_i^*(k+1) \dots \Delta u_i^*(k+N-1)\ 0\}$, yields:

$$\begin{aligned}\bar{J}_i'(k+1) = &\sum_{h=-1}^{1} \alpha_{i+h} \\ &\sum_{j=1}^{N-1} \|\hat{x}_{i+h}(k+1+j)\|^2_{\mathbb{Q}_{(i+h)j}} \\ &+ \sum_{j=0}^{N-2} \|\Delta u_i'(k+1+j)\|^2_{R_{i(j+1)}} \\ &+ \sum_{h=-1}^{1} \alpha_{i+h} \|\hat{x}_{i+h}(k+1+N)\|^2_{\mathbb{Q}_{(i+h)j}} \\ &+ \|\Delta u_i'(k+1+N-1)\|^2_{R_{iN}}.\end{aligned} \tag{4.39}$$

Assuming that $\hat{x}_{i+h}(k+N+1) = 0$, due to the zero terminal constraints, *viz.* by making $N > P = n_i$, and remarking that $\Delta u_i'(k+N) \equiv 0$ then:

$$\begin{aligned}\bar{J}_i'(k+1) = &\sum_{h=-1}^{1} \alpha_{i+h} \\ &\sum_{j=1}^{N-1} \|\hat{x}_{i+h}(k+1+j)\|^2_{\mathbb{Q}_{(i+h)j}} \\ &+ \sum_{j=0}^{N-2} \|\Delta u_i'(k+1+j)\|^2_{R_{i(j+1)}}\end{aligned} \tag{4.40}$$

or

$$\begin{aligned}\bar{J}_i'(k+1) = &\sum_{h=-1}^{1} \alpha_{i+h} \\ &\sum_{j=2}^{N} \|\hat{x}_{i+h}(k+j)\|^2_{\mathbb{Q}_{(i+h)j}} \\ &+ \sum_{j=1}^{N-1} \|\Delta u_i^*(k+j)\|^2_{R_{i(j+1)}}.\end{aligned} \tag{4.41}$$

Substituting in (4.38), it yields:

$$\begin{aligned}\bar{J}_i^*(k) = &\sum_{h=-1}^{1} \alpha_{i+h} \|\hat{x}_{i+h}(k+1)\|^2_{\bar{Q}_{(i+h)1}} \\ &+ \|\Delta u_i^*(k)\|^2_{R_{i1}} + \bar{J}'_i(k+1).\end{aligned} \tag{4.42}$$

So it is possible to conclude that $\bar{J}_i^*(k) \geq \bar{J}_i'(k+1) \geq \bar{J}_i^*(k+1)$ and consequently $\bar{J}_i^*(k)$ is a Lyapunov function for the subsystem i. Using the same methodology for the other subsystems, then the sum $\sum_i \bar{J}_i^*(k)$ is a Lyapunov function for the overall system. Using standard Lyapunov stability arguments it is proved that the closed-loop under fully-iterated distributed SIORHC is nominally asymptotically stable.

Take note that for *exact* zero terminal equality, $\forall\ i$ it implies making:

$$\hat{Y}_{Pi} \equiv \hat{X}_{Pi} = \sum_j G_{Pij}\Delta U_j + \Gamma_{Pi}x(k) = [0\ \ldots\ 0]^T \quad (C = I), \tag{4.43}$$

where in general $G_{Pij} \neq 0$, although in distributed systems it is expected that systems outside the neighborhood ($j \notin \hbar$), *i.e.* not directly connected to system i, will contribute with approximately zero to (4.43).

$$\dot{x} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 & k_1 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ k_2 & 1 & 1 \end{bmatrix} u. \tag{4.44}$$

Consider the example depicted in figure 4.3 with space state model (4.44). This example shows how to deal with unexpected interactions. Make note that the non-minimum phase zero feed forward input u_2 to system Σ_3 creating an across interaction through the input matrix B (entry 3,2). In this example, the original algorithm with $k_1 = k_2 = 0$ stabilizes the overall system because all the interactions are accounted for although the matrix B being not diagonal (as one might suppose just by looking at the block diagram). But when $k_1 \neq 0$ and $k_2 \neq 0$ this is not true anymore and (4.43) must be used to include all input contributions to the zero terminal constraints. Simulations showed that the original algorithm is able to robustly stabilize the overall system when $k_1 \neq 0$ and $k_2 = 0$, but when $k_2 > 0.005$ then (4.43) must be used. The cost function for these two cases is depicted in figures 4.4 and 4.5.

4.5 Stability analysis without full iteration

If full iteration is not possible, due to insufficient sampling time period duration (small L), and noting that:

$$\Delta U(k) \equiv \Delta U(L) = \Xi\Delta U(k-1) + \Pi x(k), \tag{4.45}$$

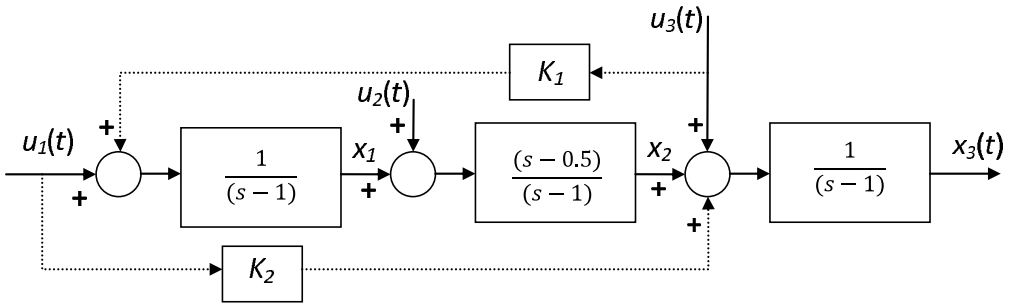

Figure 4.3: Example.

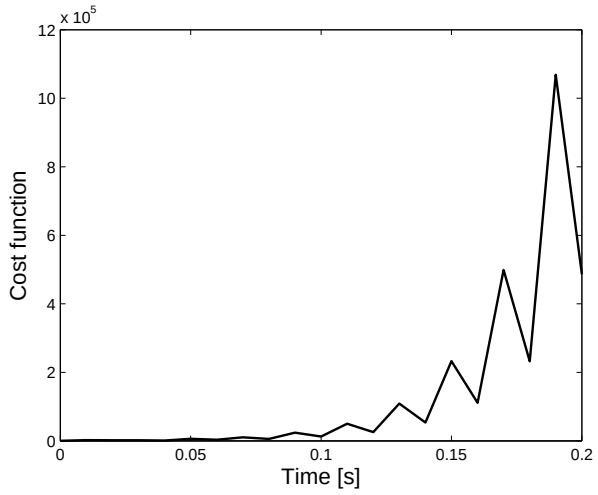

Figure 4.4: Example with the original algorithm. Cost function $k_1 = 0$, $k_2 = 0.005$.

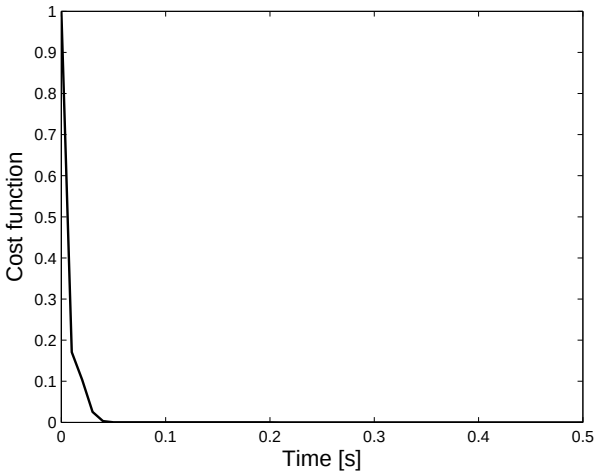

Figure 4.5: Example with modified algorithm. Cost function $k_1 = 0$, $k_2 = 0.005$.

where:

$$\Xi = (\Phi_d^{-1}\Phi_{nd})^L\Lambda, \tag{4.46}$$

$$\Pi = \sum_{l=1}^{L}(\Phi_d^{-1}\Phi_{nd})^{l-1}\Phi_d^{-1}\Upsilon \tag{4.47}$$

and noting also that $\Psi = \Upsilon x(k)$, with:

$$\Upsilon_i = -S_i \sum_{h=-1}^{1} \alpha_{i+h} G_{N(i+h)i}^T Q_{i+h}\Gamma_{Ni} + G_{Pii}^T W_i \Gamma_{Pi}, \tag{4.48}$$

the systems closed-loop dynamics is given by:

$$x(k+1) = A\ x(k) + B\ u(k) \tag{4.49}$$

$$u(k) = u(k-1) + E\ \Delta U(k) \tag{4.50}$$

$$\Delta U(k) = \Xi\ \Delta U(k-1) + \Pi\ x(k), \tag{4.51}$$

and consequently the following matrix must have its eigenvalues inside the unit circle for asymptotic stability:

$$A_{cl} = \begin{bmatrix} A & B & 0 \\ E\ \Pi A & I + E\ \Pi\ B & E\ \Xi \\ \Pi\ A & B & \Xi \end{bmatrix}. \tag{4.52}$$

Increasing L will produce a stable closed-loop since the fully-iterated stability result, stated above, must be recover for some minimum value of L.

The auxiliaries matrices Λ and E are block builded with:

$$\Lambda_i = \begin{bmatrix} 0 & I & 0 & \cdots \\ 0 & 0 & I & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & \cdots \end{bmatrix} \tag{4.53}$$

and

$$E_i = \begin{bmatrix} bidiag\ \{\ \cdots\ I_{m_i \times m_i} & 0_{m_i(N-1)\times m_i(N-1)}\ \cdots \} \end{bmatrix}, \tag{4.54}$$

respectively.

Figures 4.6-4.8 illustrate the stability result for a three unstable serially connected system with $N = 20$, $P = 3$, $L = 10$, $R = 10\ I$, $Q = I$ and $\Delta t = 0.01s$. Set-points are constant and respectively $r_1 = 1$, $r_2 = 2$ and $r_3 = 3$. Finally $y \equiv x$.

$$\dot{x} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} u. \tag{4.55}$$

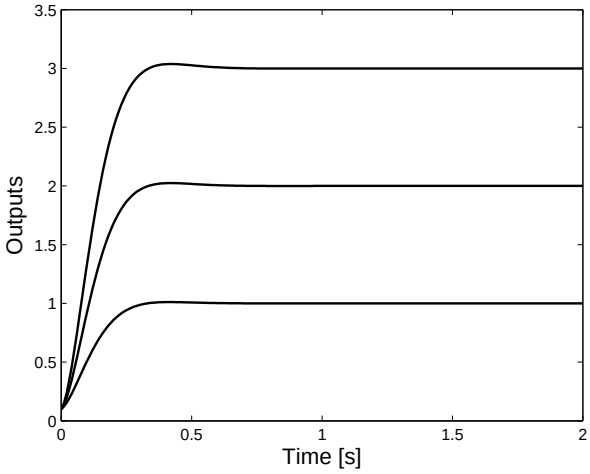

Figure 4.6: Three unstable serially connected systems; Outputs.

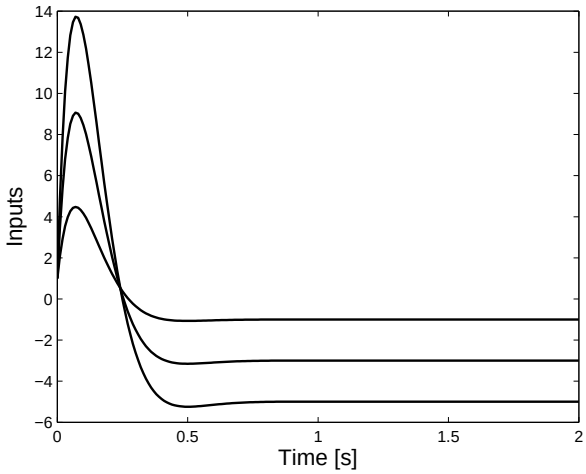

Figure 4.7: Three unstable serially connected systems; Inputs.

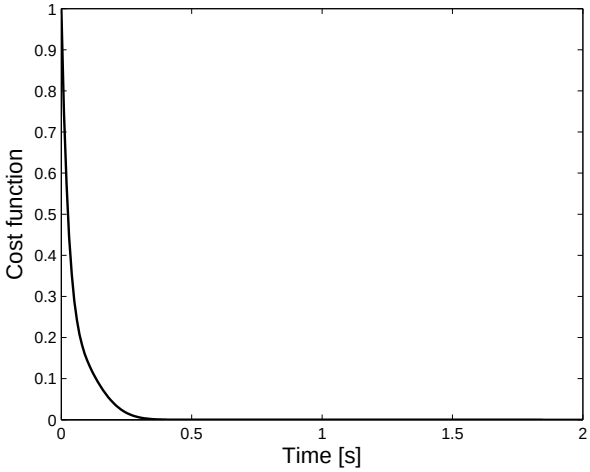

Figure 4.8: Normalized cost function; As predicted the cost function for the entire system is monotonically decreasing and therefore the system is asymptotically stable in closed-loop.

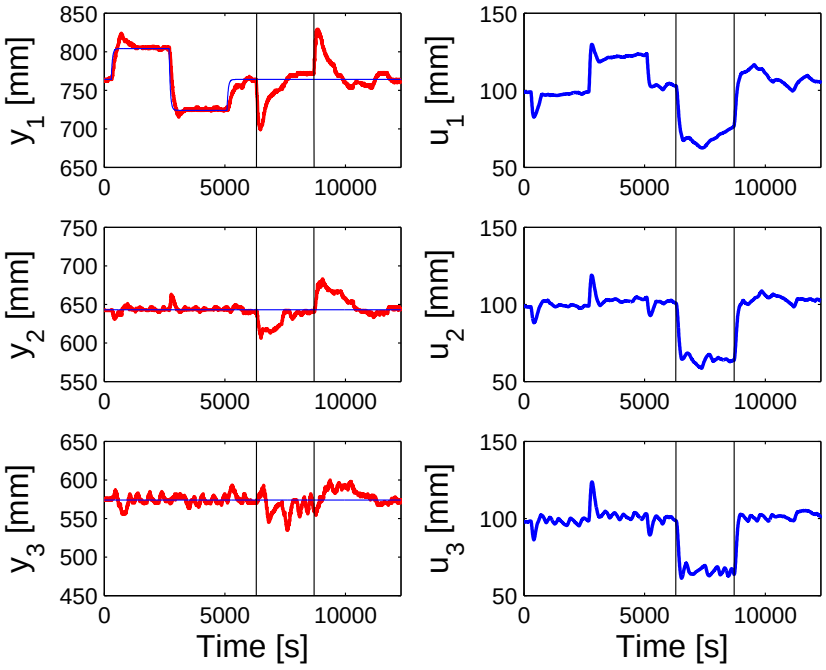

Figure 4.9: Control of a water delivery canal with Decentralized SIORHC [58]

4.6 Water delivery canal

The experimental canal of *Núcleo de Hidráulica e Controlo de Canais* (Universidade de Évora, Portugal) is taken hereafter as a prototype. The canal has four pools with a length of $35m$, separated by three undershoot gates, with the last pool ended by an overshoot gate. Four water off-takes, each one placed downstream from each branch and equipped with a valve, allow to mimic water use for irrigation. Water level sensors are installed at the end of each pool. The level sensors allow to measure values between $0\,mm$ and $900\,mm$ that corresponds to the canal border. The nomenclature is as follows: For pool number i, $i = 1, \ldots, 4$, the downstream level is denoted y_i. Pool number i ends with gate number i. The opening degree of gate i is denoted u_i. All these quantities are measured in $[mm]$.

Figure 4.9 shows experimental results obtained for upstream control of the first 3 pools using Decentralized SIORHC (dSIORHC) [58]. The vertical lines indicate the opening and closing of the off-take of pool 1.

4.7 Conclusion

A distributed version of SIORHC, a model predictive controller for linear systems with constraints on the terminal state that ensures closed-loop stability, has been presented. This results in a set of multiple local agents that act in a cooperative way to achieve a common goal - global optimal. The distributed control agents predict their control actions in a finite horizon relying in a local model that assumes only interaction with neighboring

subsystems, via the input signals. The cooperation among these agents is accomplished when using an iterative procedure that converges to the optimal decentralized (*i.e.* fully-iterated) solution. The adjustment of weights in each subsystem cost function, ranging from a so-called egocentric behavior to a pure altruistic one result in different type of closed-loop performance. The minimal requirements of a generic open computational infrastructure, where the agents can cooperate for sharing and broadcasting information, are given. Closed-loop stability results show that it can be directly applied to stable and unstable large scale linear systems. Finally, the application results obtained in a experimental water delivery canal validate this new approach to distributed control.

CHAPTER 5

MODEL PREDICTIVE CONTROL OF COUNTERCURRENT TUBULAR HEAT EXCHANGERS WITH COMPOSITE GEOMETRY

5.1 Introduction

In the previous chapters, the importance of NMPC has been shown for different prototype systems and the application of predictive and adaptive algorithms has also been accomplished. Stability is still the major issue on control, since all systems appear to have uncertainty that contributes to closed-loop instability.

In this chapter, the objective is to design both single and multi-agent predictive controllers with adaptation for a parametric uncertain transport-reaction phenomena process, described by nonlinear partial differential equations, with mass and energy transport, namely a countercurrent heat exchanger, which makes a very interesting example of a dynamically coupled system. A dynamical coupled system is a system where the states from different subsystems affect others. The purpose here is to see how the temperature of a hot and a cold fluids affect each other by means of the fluid velocities.

5.2 Prototype Model

The process considered occurs in a composite tubular geometry heat exchanger as depicted in figures 5.2 and 5.3. This geometry consists of a inner pipe inside a concentric outer pipe for flow heat exchange. The fluid in the outer pipe is countercurrent with respect to the fluid in the inner pipe. This configuration allows the study a wide variety of processes with transport phenomena as described in [20] and can be used as basis for the study of systems with more complex geometries.

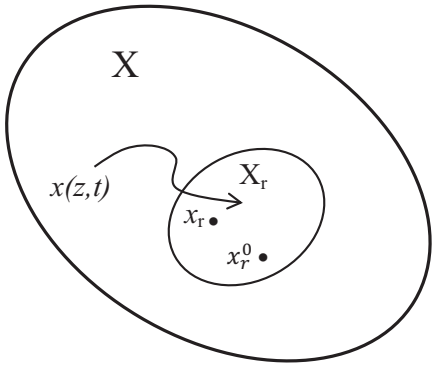

Figure 5.1: Sets with state reference curve level including real and nominal parameters.

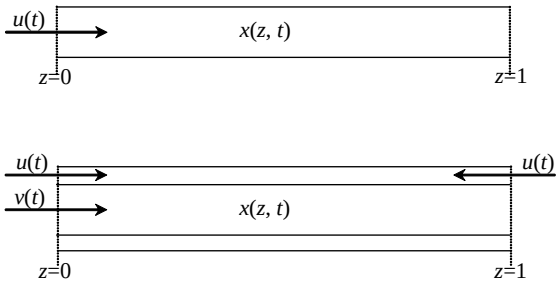

Figure 5.2: Prototype system: simple (top) and composite (bottom) geometries.

The cold fluid temperature is taken as controlled variable and hot fluid flow velocity, u, is taken as manipulated variable in single agent (or centralized) control and in u and v velocities (respectively, outer and inner fluid velocities) for multi-agent control. Furthermore, plug-flow conditions are assumed, where the fluid is radially homogeneous and the axial diffusion/dispersion is neglected and also diffusion/dispersion model are treated (see [71]).

Several studies were performed for similar heat exchanger systems as, for instance, a shell and tube heat exchanger [93], high duty counter-current heat exchangers [45] and a very close system and control type approach to this thesis work in [82]. A series of papers on countercurrent double-pipe heat exchangers has been published by [1, 2] and on heat exchanger network can be read in [9].

For better understanding, one can generalize the simple geometry system like stated in [101]:

$$-\frac{\delta_k}{L^2}\frac{\partial^2 x}{\partial z^2} + \frac{\partial x_k(z,t)}{\partial t} + \frac{v_k}{L}\frac{\partial x_k(z,t)}{\partial z} = f_k(x(z,t), d(z,t); \theta) \tag{5.1}$$

Like stated before, this set of equations models a process that occurs in a tubular system. The state variables $x_k(z,t)$ $(k = 1, \ldots, n)$ represent the fluid temperature along space ($z \in [0,1]$) and time ($t \in [0, +\infty[$) in the inner pipe, the transport velocity being v_k. From conservation principles (mass and energy balances), the system is modelled

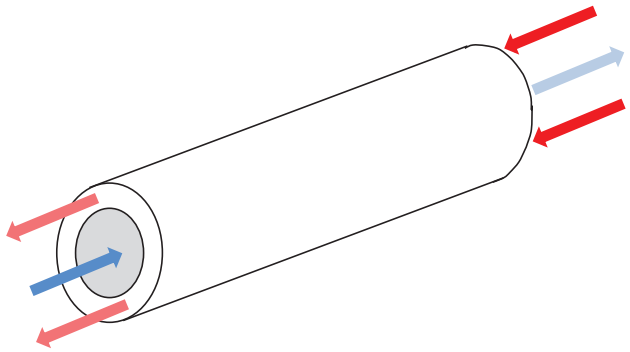

Figure 5.3: Prototype system: perspective composite geometry with countercurrent flows. Blue arrows and red arrows define cold and hot fluid flow, respectively.

by n hyperbolic PDEs in the state variables $x_k(x,t)$ and the manipulated variable is the transport velocity $u = v_k$ (except in cases in which $v_k = 0$), δ_k is a diffusion coefficient, L is the pipe length and $f_k(x, d(z,t))$ is a nonlinear bounded production term that includes possible disturbances $d(z,t)$. The uncertain parameters are put in a column vector θ and used to estimate , lying within a known set $\Theta = \{\theta : \underline{\theta} < \theta < \overline{\theta}\}$.

For the composite geometry, the following model is considered:

$$\begin{aligned} \frac{\partial w(z,t)}{\partial t} \pm \frac{u}{L}\frac{\partial w(z,t)}{\partial z} &= p(w,x,d(z,t);\theta) \\ -\frac{\delta_k}{L^2}\frac{\partial^2 x}{\partial z^2} + \frac{\partial x_k(z,t)}{\partial t} + \frac{v_k}{L}\frac{\partial x_k(z,t)}{\partial z} &= f_k(w,x;\theta) \end{aligned} \tag{5.2}$$

The equations (5.2) present an additional temperature $w(z,t)$ balance in the outer pipe, being u the fluid velocity in the same pipe. Vector θ again denotes the vector of uncertain parameters to estimate.

The objective is to control the output assumed to be a smooth nonlinear state function weighted over space, by manipulating the fluid velocity $u(t)$.

5.3 NMPC Nominal Formulation and Adaptive Control

The nominal formulation as well as stability for this class of systems is thoroughly discussed in previous chapters. The controller design for the heat exchanger here presented is solely based on the stability conditions and adaptation, so the focus will be on a more effective discussion of the adaptation mechanism to be combined with the proposed NMPC.

5.3.1 Adaptation with parameters observer

The proposed way to deal with parameter uncertainty is to use an observer for estimating parameters. Considering the state to be accessible as described in [4, 29, 56] and considering

the full model (5.2) in the following matrix form:

$$\frac{\partial \eta}{\partial t} = \begin{bmatrix} p(\eta, d) \\ f(\eta) + \frac{\mathcal{D}}{L^2}\frac{\partial^2 x}{\partial z^2} - \frac{A_v}{L}\frac{\partial x}{\partial z} \end{bmatrix} \pm \begin{bmatrix} \frac{\partial w}{\partial z} \\ 0 \end{bmatrix} \frac{u}{L} \tag{5.3}$$

where $\eta = [w \;\; x^T]^T$.

It is also considered that the output does not explicitly depends on w, $\frac{dh}{d\eta} = \begin{bmatrix} 0 & \frac{dh}{dx} \end{bmatrix}$ and assumed that typical process production terms are linear in θ:

$$f(\eta;\theta) = f_0(\eta) + \sum_{i=1}^{q} \theta_i f_i(\eta) = f_0(\eta) + F(\eta)\theta \tag{5.4}$$

$$p(\eta;\theta) = p_0(\eta) + \sum_{i=1}^{q} \theta_i p_i(\eta) = p_0(\eta) + P(\eta)\theta \tag{5.5}$$

it yields:

$$\frac{\partial \eta}{\partial t} = \begin{bmatrix} 0 \\ \mathcal{L}x \end{bmatrix} \pm \begin{bmatrix} \frac{\partial w}{\partial z} \\ 0 \end{bmatrix} \frac{u}{L} + \begin{bmatrix} p_0(\eta) \\ f_0(\eta) \end{bmatrix} + \begin{bmatrix} P(\eta) \\ F(\eta) \end{bmatrix} \theta \tag{5.6}$$

where:

$$\mathcal{L}x = \frac{\mathcal{D}}{L^2}\frac{\partial^2 x}{\partial z^2} - \frac{A_v}{L}\frac{\partial x}{\partial z} \tag{5.7}$$

Writing the observer filter dynamics in the form [4, 114]:

$$\begin{aligned} \frac{\partial \eta_a}{\partial t} &= \begin{bmatrix} 0 \\ \mathcal{L}x \end{bmatrix} \pm \begin{bmatrix} \frac{\partial w}{\partial z} \\ 0 \end{bmatrix} \frac{u}{L} + \begin{bmatrix} p_0(\eta) \\ f_0(\eta) \end{bmatrix} \\ &+ \begin{bmatrix} P(\eta) \\ F(\eta) \end{bmatrix} \hat{\theta} + \mathcal{K}(\eta - \eta_a) \end{aligned} \tag{5.8}$$

where $\mathcal{K} > 0$, the observation error dynamics ($e_a = \eta - \eta_a$) is given by:

$$\frac{\partial e_a}{\partial t} = \begin{bmatrix} P(\eta) \\ F(\eta) \end{bmatrix} \tilde{\theta} - \mathcal{K} e_a \tag{5.9}$$

Considering the Lyapunov candidate function [67]:

$$V(e_a, \tilde{\theta}) = \frac{1}{2}\left(\int_0^1 e_a^T e_a \, dz + \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \right) \tag{5.10}$$

Differentiating V with respect to time, using the error dynamics and choosing:

$$\dot{\hat{\theta}} = -\Gamma \int_0^1 \begin{bmatrix} P(\eta) \\ F(\eta) \end{bmatrix}^T e_a \, dz \tag{5.11}$$

it guarantees:

$$\dot{V}(e_a) < 0; \quad e_a \neq 0 \tag{5.12}$$

making the errors go to zero ($e_a(t)$ and $\tilde{\theta} \rightarrow 0$ as $t \rightarrow \infty$), after initial conditions transient ($w(0),\ x(0)$ and $\hat{\theta}(0)$), if the nonlinear functions matrix:

$$\left[\begin{array}{c} P(\eta) \\ F(\eta) \end{array} \right] \neq 0 \tag{5.13}$$

locally, as a way to assure that the adaptation error goes to zero.

Note that $\tilde{\theta}$ can be bounded by the use of the projection method, as shown earlier and thereby constraining the estimated to the interior of a bounded convex parameters space. The parameters projection uses the following rule (see [66]):

$$\text{Proj}\left\{\tau, (\overline{\theta}, \underline{\theta}, \epsilon)\right\} = \begin{cases} max(0, \frac{\epsilon - \hat{\theta} + \overline{\theta}}{\epsilon})\tau & \hat{\theta} \geq \overline{\theta} \text{ and } \tau > 0 \\ max(0, \frac{\epsilon + \hat{\theta} - \underline{\theta}}{\epsilon})\tau & \hat{\theta} \leq \underline{\theta} \text{ and } \tau < 0 \\ \tau & otherwise \end{cases} \tag{5.14}$$

with $\overline{\theta} + \epsilon \geq \hat{\theta} \geq \underline{\theta} - \epsilon$ and where τ is the update law.

5.3.2 Countercurrent Heat Exchanger Model

Consider the model of a countercurrent heat exchanger:

$$\frac{\partial w}{\partial t} - \frac{u}{L}\frac{\partial w}{\partial z} = b(x - w) \tag{5.15}$$

$$\frac{\partial x}{\partial t} + \frac{v}{L}\frac{\partial x}{\partial z} = a(w - x) \tag{5.16}$$

in which w and x are respectively the outer and inner pipe fluid temperature, a and b the outer and inner exchange coefficient [min^{-1}], v is the inner pipe fluid velocity [$m\ min^{-1}$] and the manipulated variable is the outer pipe fluid velocity u.

In the adaptive case and assuming the certain equivalence, the parameters are replaced by the estimates:

$$\begin{aligned} \dot{\hat{a}} &= -\gamma_a(w(1,t) - x(1,t))(x(1,t) - \eta_a) \\ \dot{\hat{b}} &= -\gamma_b(w(0,t) - x(0,t))(w(0,t) - \eta_b) \end{aligned} \tag{5.17}$$

with local observation dynamics, respectively:

$$\dot{\eta}_a = -\frac{v}{L}\frac{\partial x}{\partial z} \mid_{z=1} + \hat{a}\ (w(1,t) - x(1,t)) + \kappa_a(x(1,t) - \eta_a) \tag{5.18}$$

$$\dot{\eta}_b = \frac{u}{L}\frac{\partial w}{\partial z} \mid_{z=0} + \hat{b}\ (x(0,t) - w(0,t)) + \kappa_b(x(0,t) - \eta_b) \tag{5.19}$$

Figure 5.4 shows the parameters convergence, with $\hat{a}(0) = \hat{b}(0) = 0.5$ and $a = b = 0.2$ for the true values and figure 5.5 shows additionally a 20% change in parameter a, meaning $a(t > 120) = 0.25$. This simulations uses 200 space elements and values from tables 5.1 and 5.2.

Table 5.1: Heat Exchanger parameters, initial and boundary conditions.

Parameter	**Value**	**Units**
L	1	m
a	0.2	min^{-1}
b	0.2	min^{-1}
$w(z,0)$	50.0	$^{\circ}C$
$x(z,0)$	25.0	$^{\circ}C$
$w(1,t)$	50.0	$^{\circ}C$
$x(1,t)$	25.0	$^{\circ}C$

Table 5.2: Controller parameters.

Parameters	**Values**
γ_a	0.002
γ_b	0.002
κ_a	0.500
κ_b	0.500

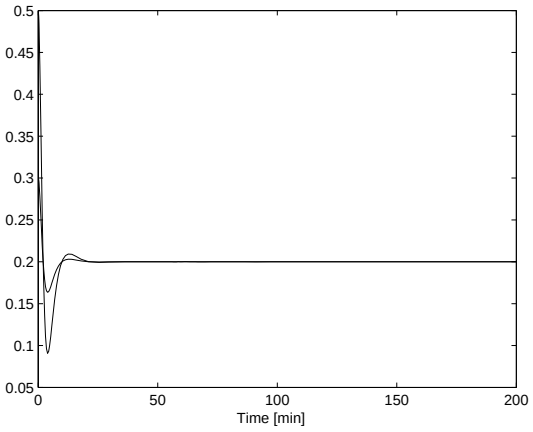

Figure 5.4: Parameters estimation: $\hat{a}$, $\hat{b}$ $[min^{-1}]$.

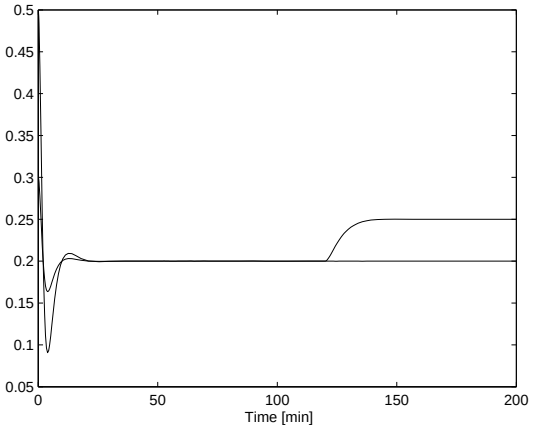

Figure 5.5: Parameters estimation for disturbed system: $\hat{a}$, $\hat{b}$ $[min^{-1}]$.

5.3.3 NMPC combination with LAL dynamics

In the adaptive NMPC for the SISO case (present in decentralized or distributed control), the RHC+LAL can be formulated as:

$$\min_{u} J = \int_{t}^{t+T} \left(\tilde{y}^2 + \rho\tilde{u}^2\right)\ d\tau \tag{5.20}$$

subject to the predictive model (5.2) in which $\theta = \hat{\theta}(t)$ and also to input, state, output constraints:

$$\mathcal{C}(\dot{\xi}(t), \dot{y}(t), \dot{u}(t), \xi(t), y(t), u(t), t) \leq 0 \tag{5.21}$$

and the stability constraint:

$$\mathcal{S}(\xi(t), u(t); \overline{\theta}, \underline{\theta}) \leq 0 \tag{5.22}$$

and the additional distributed observer and parameters estimate dynamics:

$$\frac{\partial \eta_a}{\partial t} = \mathcal{L}_\eta + \varphi_0(\eta) + \varphi(\eta)\hat{\theta} - \mathcal{K}(\eta - \eta_a) \tag{5.23}$$

$$\dot{\hat{\theta}} = \text{Proj}\left\{\Gamma \int_0^1 \varphi^T(\eta)(\eta - \eta_a)\ dz, (\overline{\theta}, \underline{\theta}, \epsilon)\right\}, \tag{5.24}$$

where η and η_a are the system and observer state respectively, $\mathcal{L}_\eta$ is the space operator on the systems state, containing convection and diffusion terms, and $\varphi_0(\eta)$ and $\varphi(\eta)$ are obtained from the production terms $f(x;\theta)$ and $p(w;\theta)$. Note that the parameters projection ensures that the estimates stay inside the bounded meaningful physical set Θ and the stability condition must robustly stabilize the system within that set.

In the centralized case, corresponding to the MIMO case, the formulation RHC+LAL uses the set of equations (5.25):

$$\begin{aligned} \min_{u(\bar{t})} J &= \int_{t}^{t+\mathcal{T}} \left(\left(\|\tilde{y}\|_Q^2 + \|\tilde{u}\|_R^2\right)\right)\ d\tau \\ s.\,t. \quad & \mathcal{O}(\xi(x(\bar{t})), y(\bar{t}), u(\bar{t}), \bar{t}) \leq 0 \\ & \max_{\theta\in\Theta}\{\dot{V}(\tilde{x}(t);\theta)\} + \alpha V(\tilde{x}(t)) < 0 \\ & u(\bar{t}) = seq\{u_1, \ldots, u_{N_u}\} \end{aligned} \tag{5.25}$$

5.4 Simulation results

Several simulations were performed in order to test the controller design. The first three simulation scenarios for centralized (or single agent) control with constant inner pipe velocity. The next set of simulations demonstrate a fully centralized architecture that include both fluid velocities. Finally, a new set of simulations were carried out in order to test a noncooperative distributed architecture (different controllers with different control

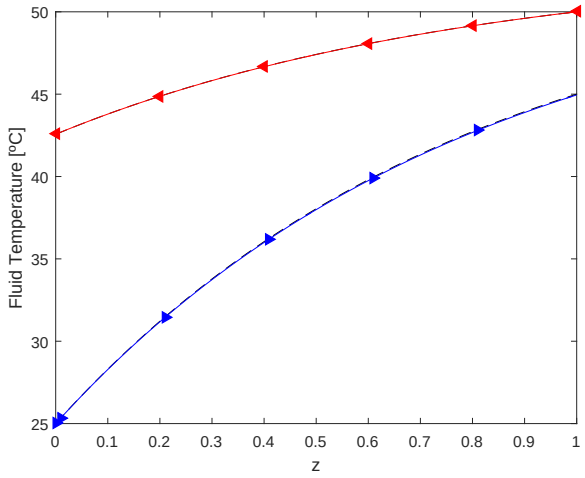

Figure 5.6: Simulation 1: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

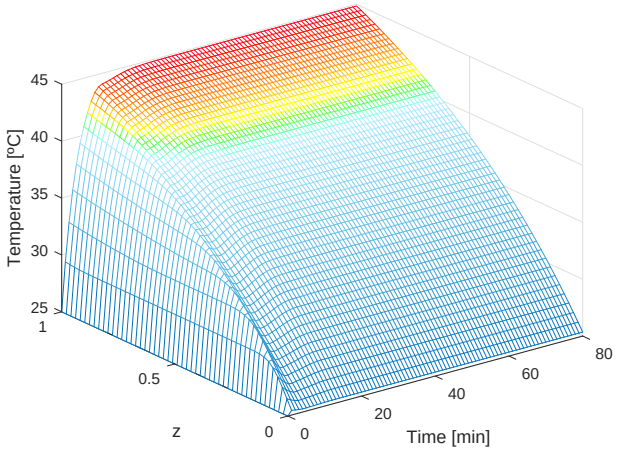

Figure 5.7: Simulation 1: Cold fluid temperature profile.

action weights) to test the performance and robustness of all algorithms. All simulations use 100 state equations and use the same initial conditions.

5.4.1 Centralized control

5.4.1.1 Simulation 1: constant inner fluid velocity

In simulation 1 the controller action is on hot fluid velocity for a steady-state velocity $u = 0.27\ m/min$ and a constant cold fluid velocity of $v = 0.1\ m/min$ as shown in figures 5.6- 5.9 The chosen control penalty weight is 0.5 and the rest of the parameters are described in tables 5.1 and 5.2.

As can be seen in figure 5.6, both temperatures track the reference with almost pinpoint

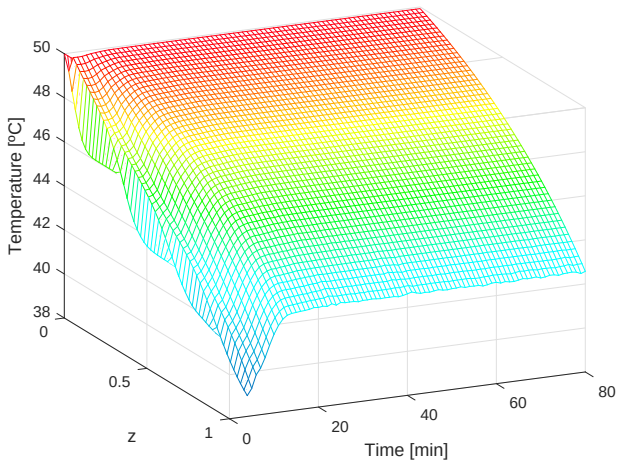

Figure 5.8: Simulation 1: Hot fluid temperature profile.

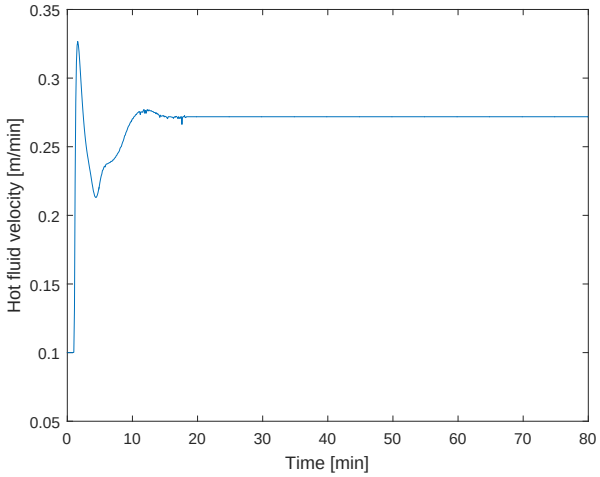

Figure 5.9: Simulation 1: Hot fluid velocity [m/min].

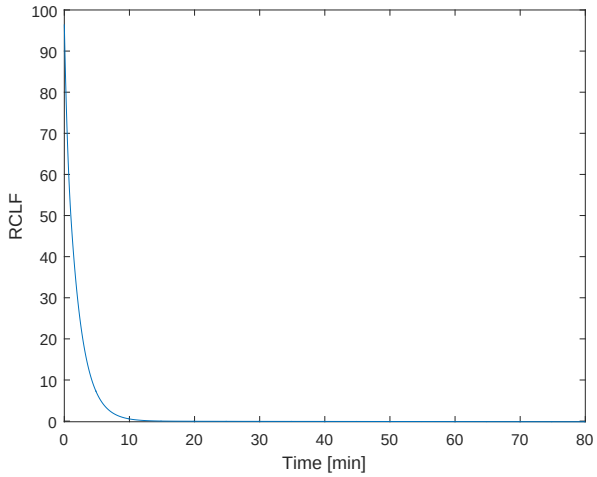

Figure 5.10: Simulation 1: Robust control Lyapunov function.

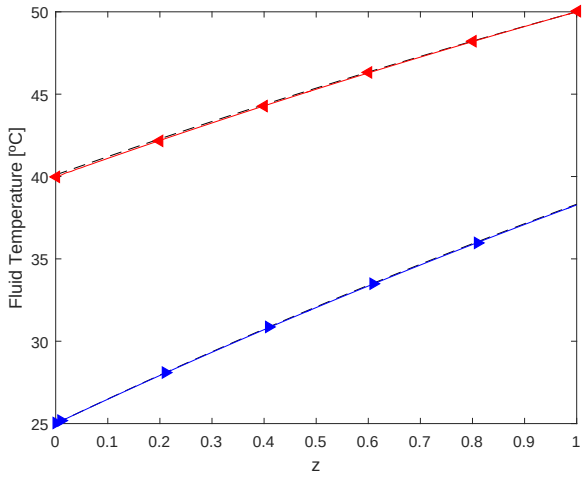

Figure 5.11: Simulation 2: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

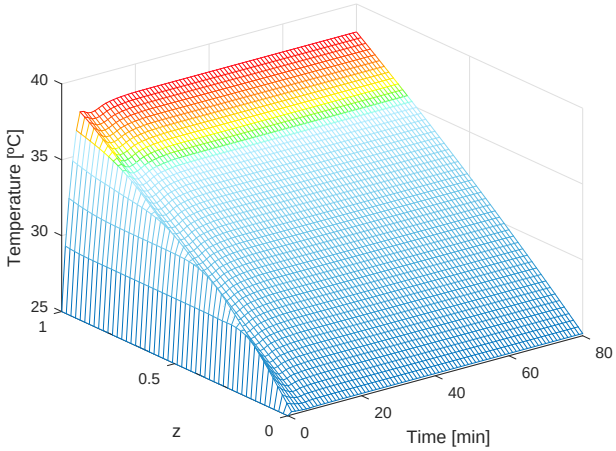

Figure 5.12: Simulation 2: Cold fluid temperature profile.

accuracy, resulting in state profiles as depicted in figures 5.7 and 5.8 for cold and hot fluid temperatures, respectively. The outer fluid velocity is shown in figure 5.9. Figure 5.10 shows the robust Lyapunov control function continuously decreasing.

5.4.1.2 Simulation 2: doubling inner fluid velocity

For simulation 2 the inner fluid velocity was doubled to $v = 0.2\ m/min$ maintaining the rest of the parameters unchanged for sake of comparison with simulation 1. Figure 5.11 shows the new references on the heat exchange process.

As can be seen, references changed due to different velocities, but still the controller maintained the ability to track the references. Figures 5.12 and 5.13 depict the new temperature profiles and figures 5.14 and 5.15 show, respectively the velocity profile and the

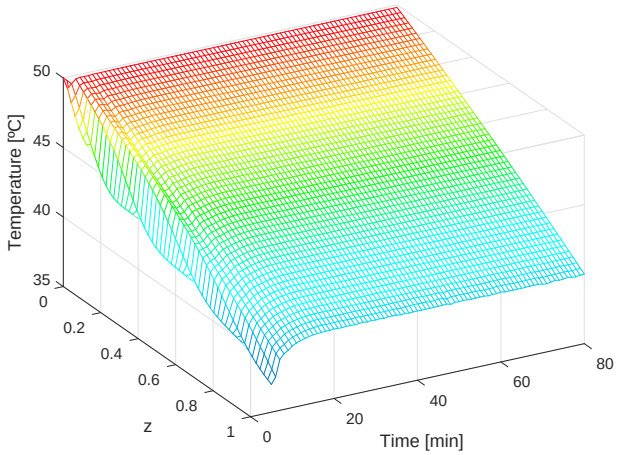

Figure 5.13: Simulation 2: Hot fluid temperature profile.

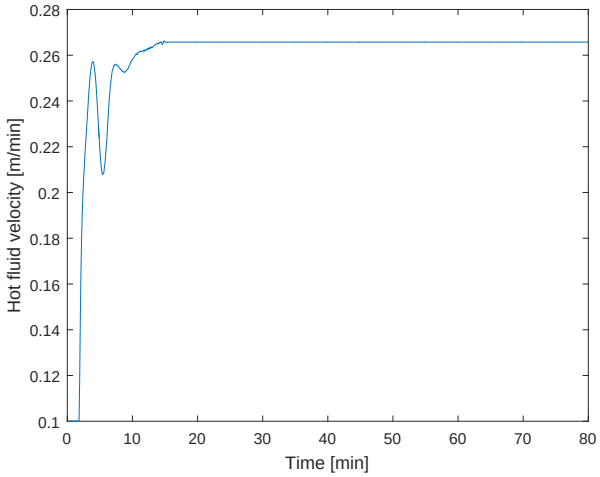

Figure 5.14: Simulation 2: Hot fluid velocity [m/min].

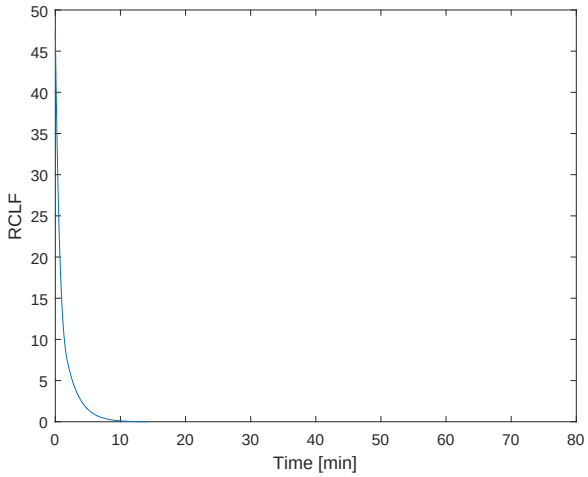

Figure 5.15: Simulation 2: Robust control Lyapunov function.

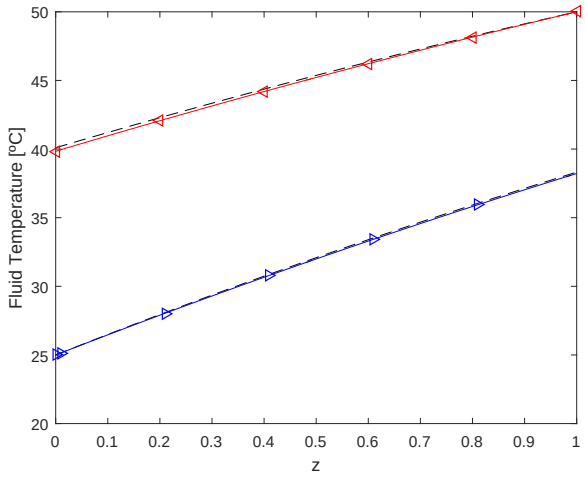

Figure 5.16: Simulation 3: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

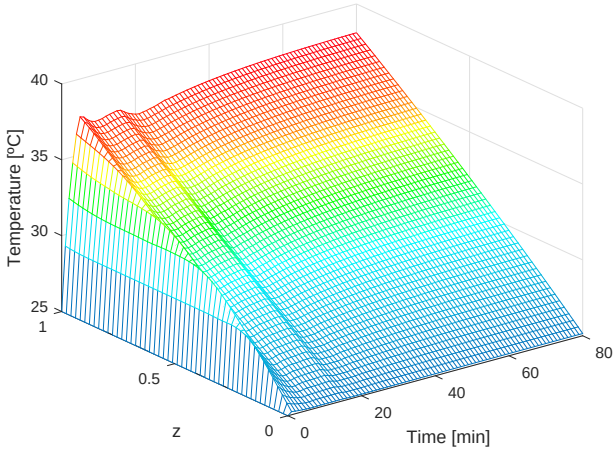

Figure 5.17: Simulation 3: Cold fluid temperature profile.

decreasing Lyapunov function.

5.4.1.3 Simulation 3: Fault-tolerance testing

In simulation 3, using the scenario values from the previous simulation, the controller was subject to a flaw in the velocity of the outer pipe fluid. This intends to simulate what happens to the temperature control if the pump for the hot fluid works intermittently or is faulty.

Focusing on the input and RCLF (figures 5.19 -5.20), the controller shows excellent performance even when subject to a fault in the hot fluid velocity at $t \in [15, 17]$ min. This faulty movement of the hot fluid is also shown in figures 5.17 and 5.18, as well as in 5.16 where it is denoted a slight deviation from the reference. A closer analysis of the Lyapunov

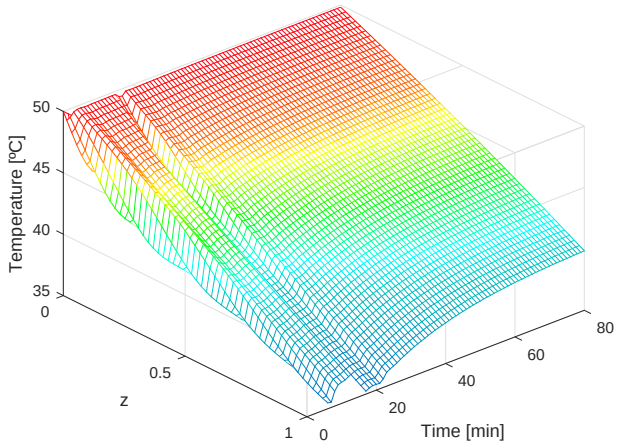

Figure 5.18: Simulation 3: Hot fluid temperature profile.

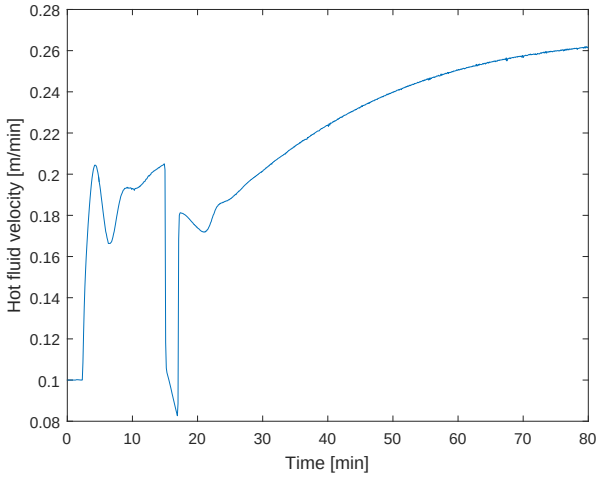

Figure 5.19: Simulation 3: Hot fluid velocity [m/min].

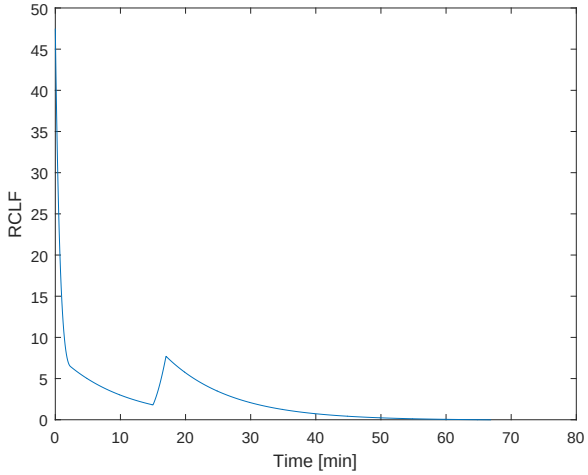

Figure 5.20: Simulation 3: Robust control Lyapunov function.

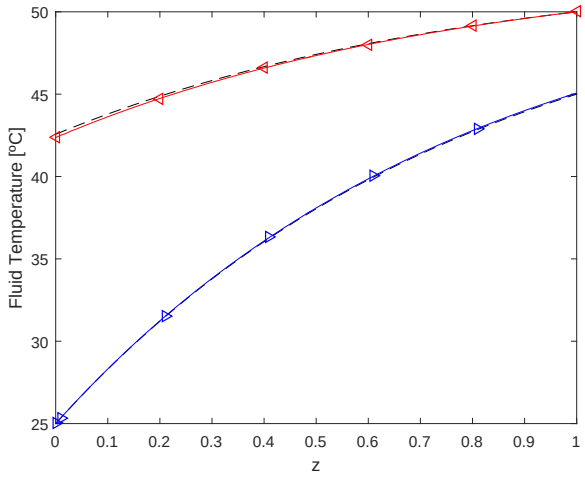

Figure 5.21: Simulation 4: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

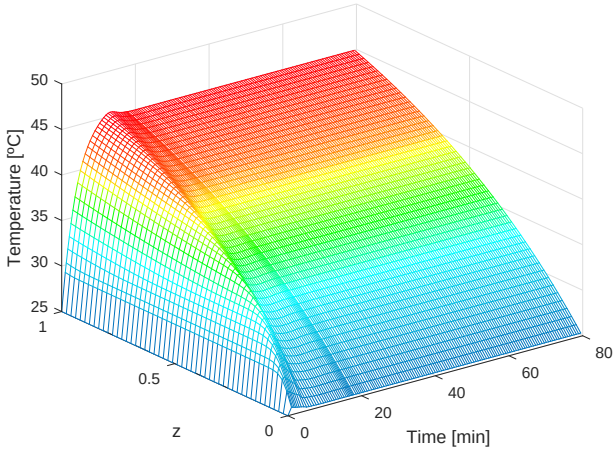

Figure 5.22: Simulation 4: Cold fluid temperature profile.

control function, one can see that the fail in u promotes an increase in the RCLF. As soon has the pump is fixed, the controller increases the velocity in order to track the steady-state reference. In this particular case, the time span seems to be short, but still is possible to see that the adaptation combined with stable NMPC maintain the integrity and robustness of the projected controller.

5.4.1.4 Simulations 4, 5 and 6: Fully centralized architecture

In simulations 4, 5 and 6, the controller has been redesigned to incorporate both hot and cold fluid velocities in the control action. The main intention is to serve as a comparison to the simulation results of the first three tests.

From figures 5.21 -5.25 (referred to simulation 4) analysis one can conclude that the

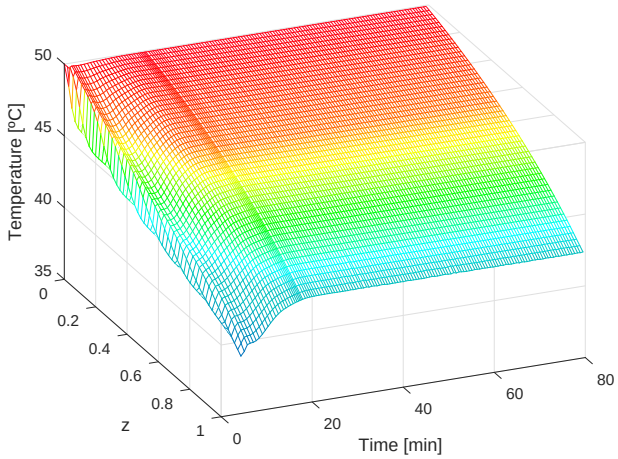

Figure 5.23: Simulation 4: Hot fluid temperature profile.

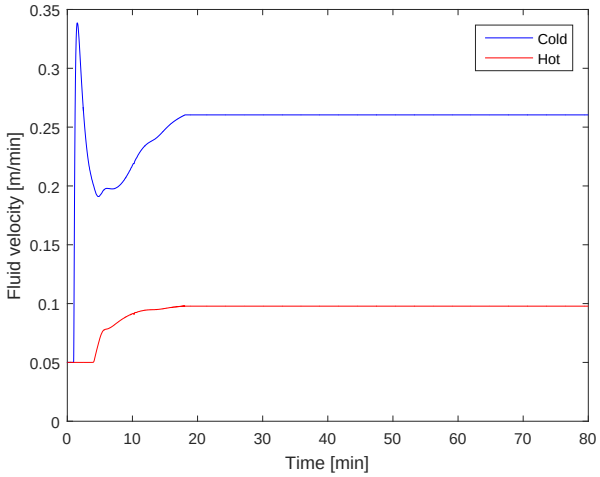

Figure 5.24: Simulation 4: Hot and cold fluid velocity [m/min].

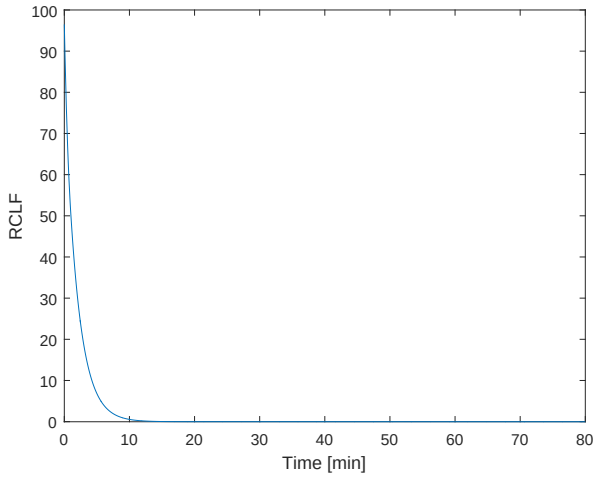

Figure 5.25: Simulation 4: Robust control Lyapunov function.

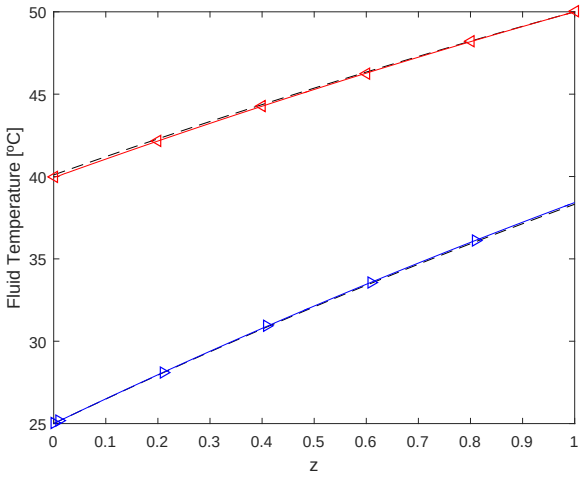

Figure 5.26: Simulation 5: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

controller works as well as the original version, especially when dealing with the objective output. Figure 5.24 shows the inclusion of the now controlled cold fluid velocity. In terms of computational effort, this new controller does decrease the time for simulation (approximately $5 - 10\%$).

In simulation 5 both scenarios from simulations 2 and 3 are combined. It has been included a reduction of the outer fluid velocity between $t \in [15, 17]$ min for a $v = 0.2$ m/min. Figures 5.26 to 5.30 show that the fully centralized version takes advantage of actuating over both velocities to compensate the fail on one of the velocities. The control input signal (figure 5.29) shows that a smaller weight on control action would be better to prevent sudden changes, although adaptation seems to be sufficient enough to prevent degrading the performance. RCLF (figure 5.30) shows a slight increase in the Lyapunov control function for the fault in u.

Finally, figures 5.31 -5.35 show the effect of larger cumulative fail in both fluid velocities. Even in this scenario, the controller keeps integrity and although both temperature profiles present bigger overshoots (positive in the cold fluid and negative in the hot fluid, in respect to the reference of each temperature). Computational effort is still smaller in this fully centralized version than in the one presented in the first three simulations.

5.4.2 Non-cooperative Distributed control

In this section, the control system has been again redesigned. This time, a non-cooperative distributed architecture has been adopted, where both controllers change information about their local optimizations in a non-iterative parallel fashion, as described in chapter 1, section 1.3. The main objective here is to evaluate how different architectures may differ in this type of uncertain nonlinear system. Both controllers target for closed-loop stability.

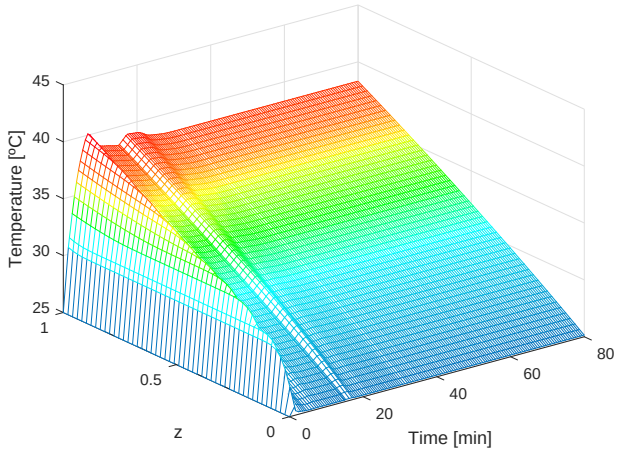

Figure 5.27: Simulation 5: Cold fluid temperature profile.

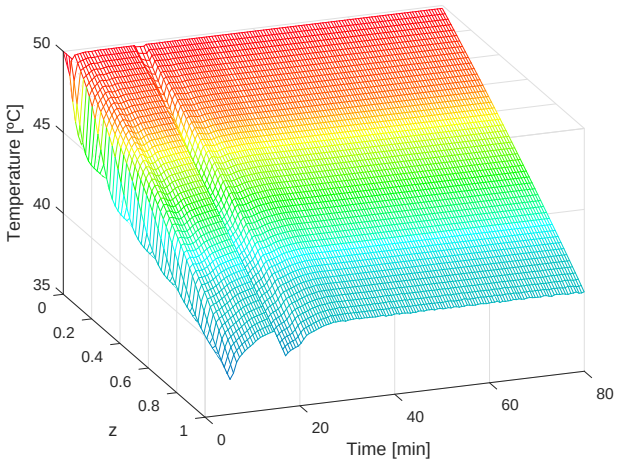

Figure 5.28: Simulation 5: Hot fluid temperature profile.

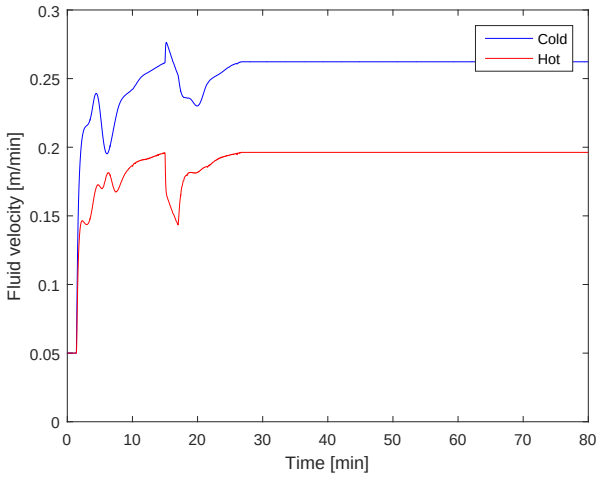

Figure 5.29: Simulation 5: Hot and cold fluid velocity [m/min].

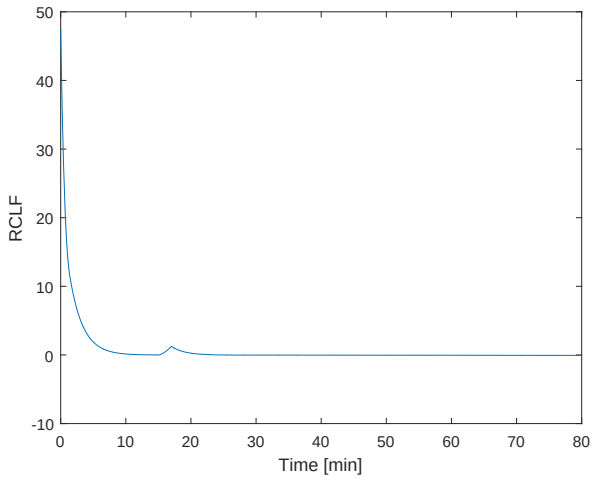

Figure 5.30: Simulation 5: Robust control Lyapunov function.

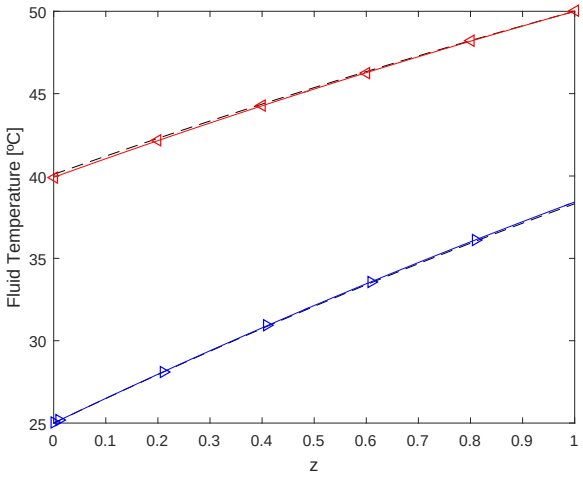

Figure 5.31: Simulation 6: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

5.4.2.1 Simulation 7: different control weights

In this simulation controllers use different control weights, α and fluid velocities are kept close at $u = 0.2\ m/min$ and $v = 0.201\ m/min$. Physically speaking, both velocities in a heat exchange installation can be equal, but numerically making both velocities equal leads to a indetermination, creating instability. Control weight for the controller on outer fluid flow velocity, $\alpha_1 = 0.5$ and for the inner fluid flow, $\alpha_2 = 0.1$. This prevents both controllers to compete in order to obtain better performance over each other. Signal saturation is also avoided.

Figures 5.36 to 5.40 depict this simulation results. Computational effort has increased due to the need to maintain control integrity in both subsystems.

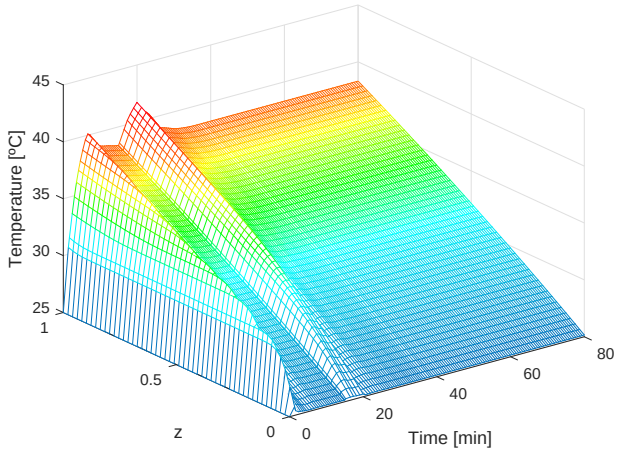

Figure 5.32: Simulation 6: Cold fluid temperature profile.

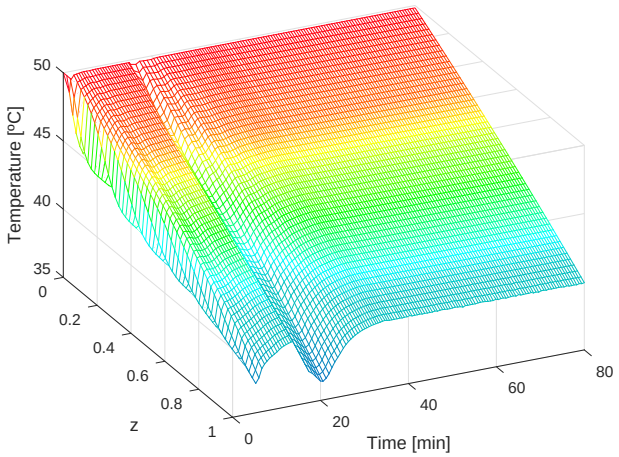

Figure 5.33: Simulation 6: Hot fluid temperature profile.

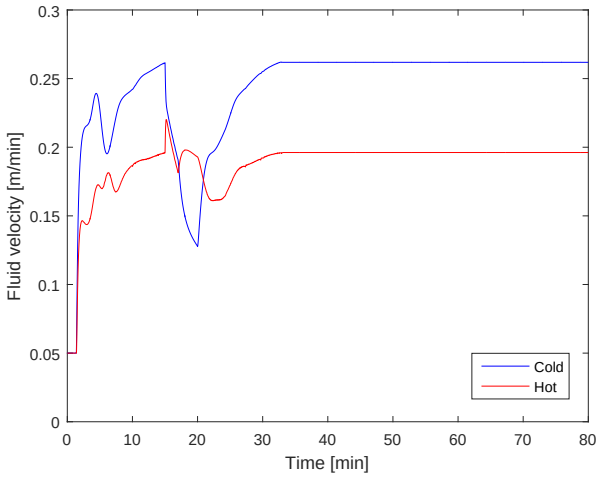

Figure 5.34: Simulation 6: Hot fluid velocity [m/min].

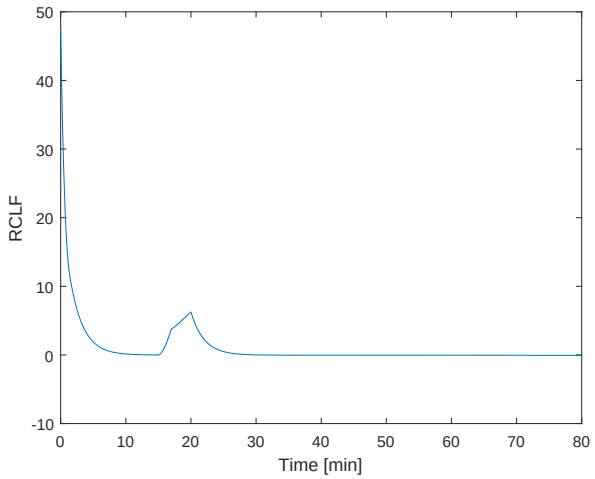

Figure 5.35: Simulation 6: Robust control Lyapunov function.

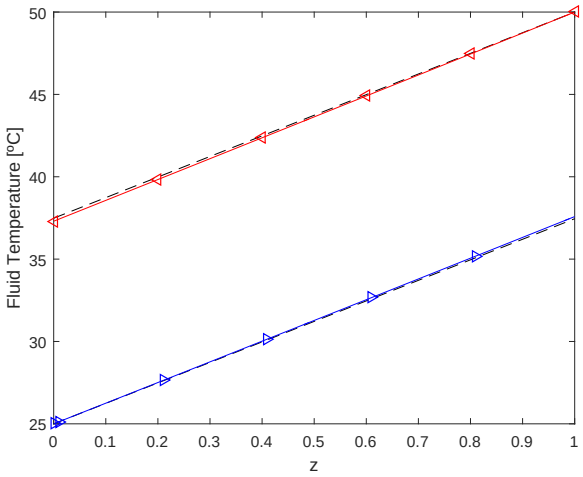

Figure 5.36: Simulation 7: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

5.4.2.2 Simulations 8 and 9: fault-tolerance scenarios

This last two simulations intend to show how stability and performance are affected with simple and cumulative fails in the feeding fluids for non-cooperative DMPC.

Figures 5.41 to 5.45 depict this simulation 8 results. Computational effort has, as expected, increased due to the need to maintain control integrity in both subsystems. Figures 5.46 to 5.50 show the results for simulation 9, where velocities are $u = 0.27\ m/min$ and $v = 0.3\ m/min$. Both controllers are subject to sudden decreases in v and v for a certain amount of time, namely $t \in [15, 17]\ min$ and $t \in [15, 20]\ min$ respectively. Once again both controllers prove to be robust even in worst-case scenarios.

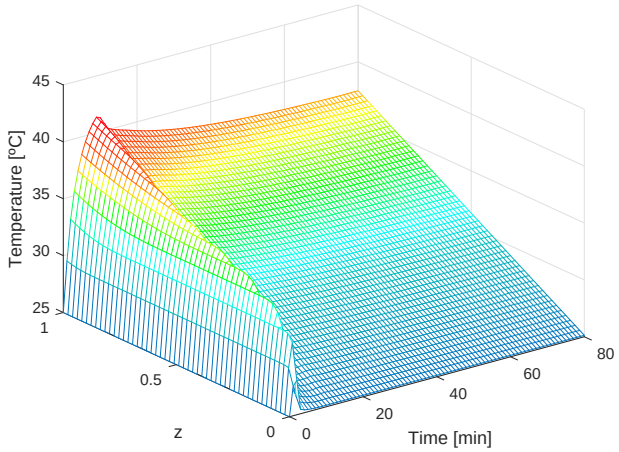

Figure 5.37: Simulation 7: Cold fluid temperature profile.

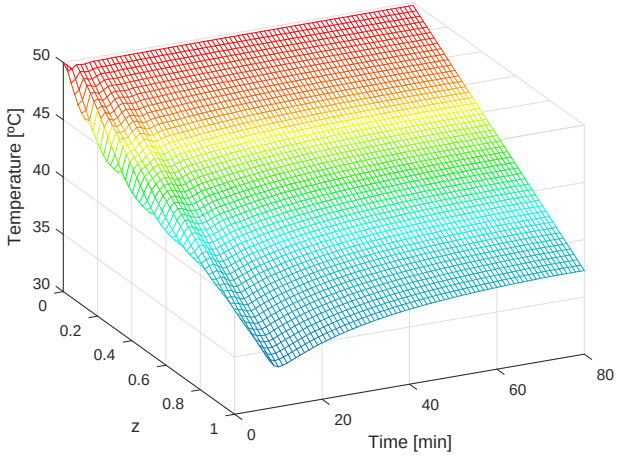

Figure 5.38: Simulation 7: Hot fluid temperature profile.

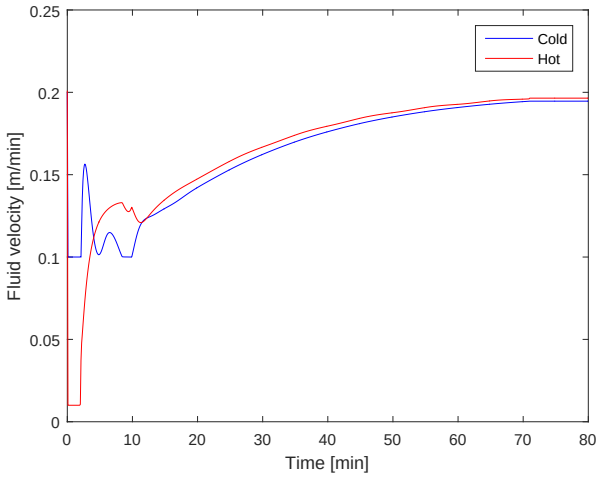

Figure 5.39: Simulation 7: Hot and cold fluid velocity [m/min].

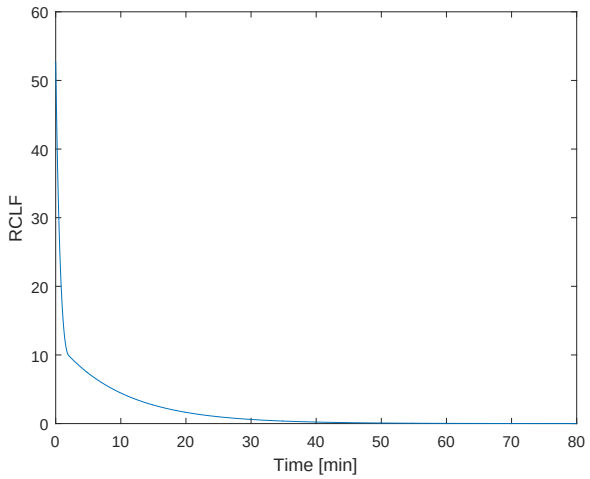

Figure 5.40: Simulation 7: Robust control Lyapunov function.

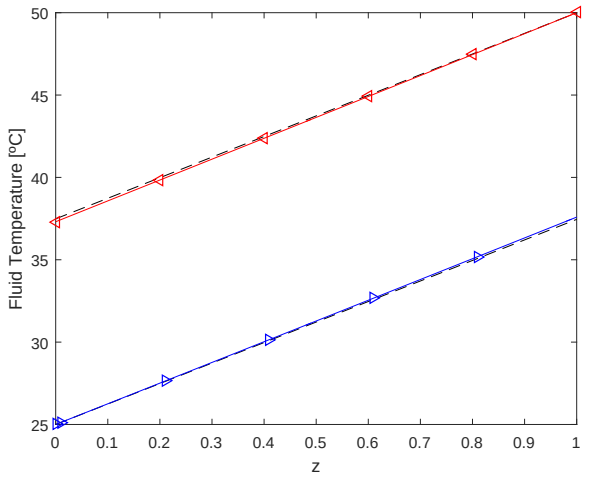

Figure 5.41: Simulation 8: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

5.5 Conclusions

An adaptive NMPC formulation for an uncertain class of distributed parameter systems is introduced and exemplified. Different control architectures have been developed and successfully applied to a countercurrent heat exchanger, a thorough dynamical coupled system. Low computational effort for on-line parameter estimation as a way to deal with uncertainty is shown to justify this type of alternative approach, especially in distributed control architectures. Controller performance and robustness have been tested in different scenarios as a way to show the importance of stability conditions combined with

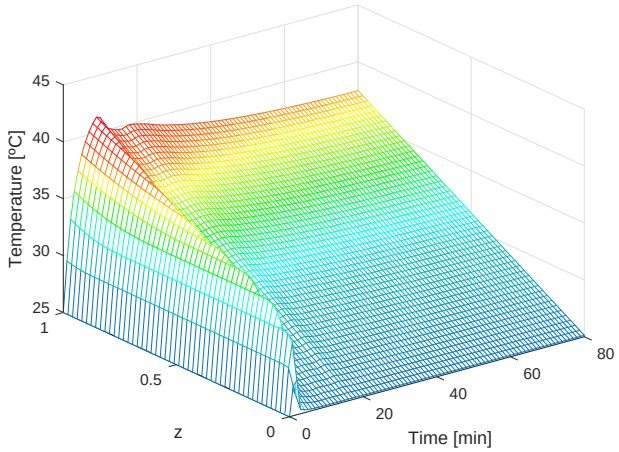

Figure 5.42: Simulation 8: Cold fluid temperature profile.

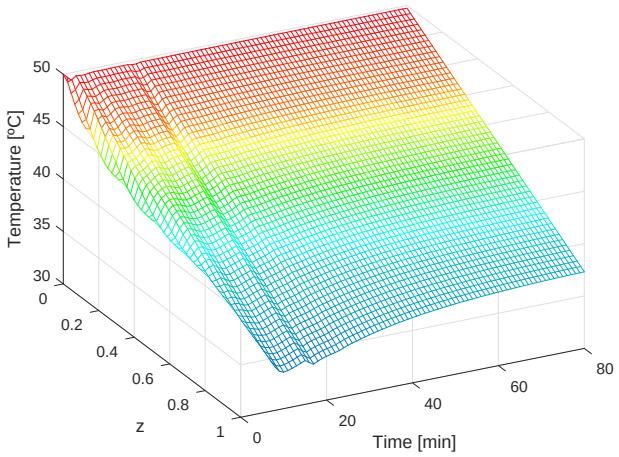

Figure 5.43: Simulation 8: Hot fluid temperature profile.

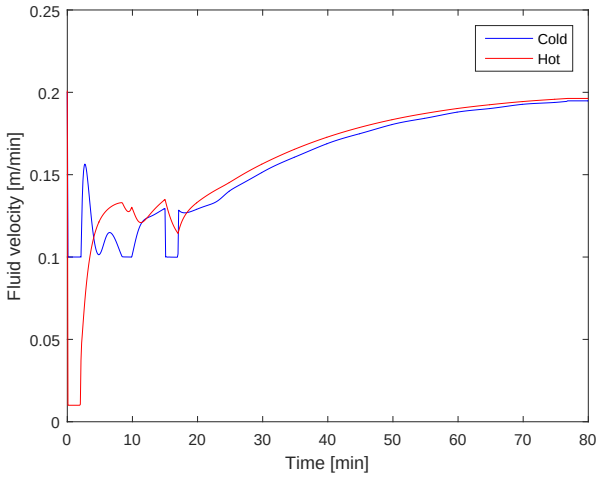

Figure 5.44: Simulation 8: Hot fluid velocity [m/min].

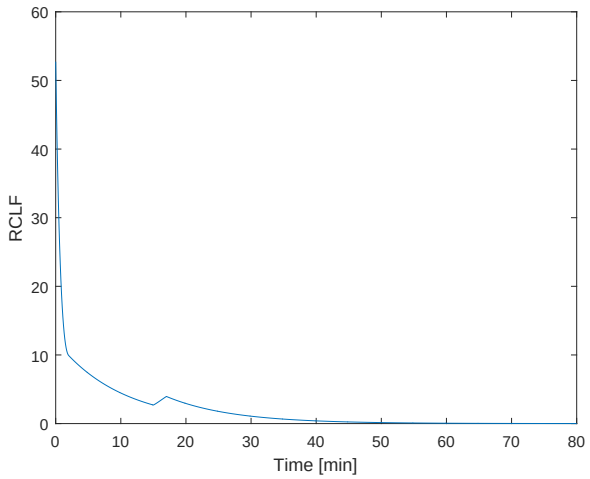

Figure 5.45: Simulation 8: Robust control Lyapunov function.

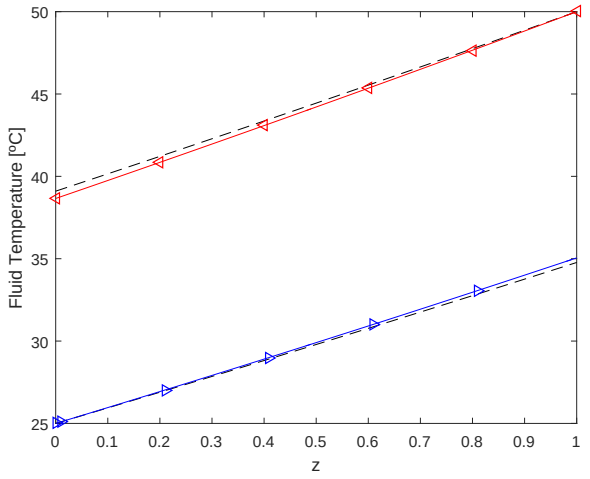

Figure 5.46: Simulation 9: Temperature tracking reference. Dotted lines mark both references; Cold fluid temperature (blue line with left-to-right direction marker) and hot fluid temperature (red line with right-to-left direction marker).

adaptation for SISO and MIMO systems.

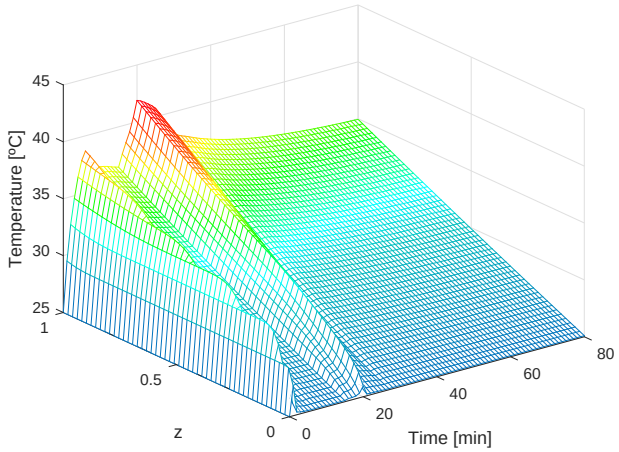

Figure 5.47: Simulation 9: Cold fluid temperature profile.

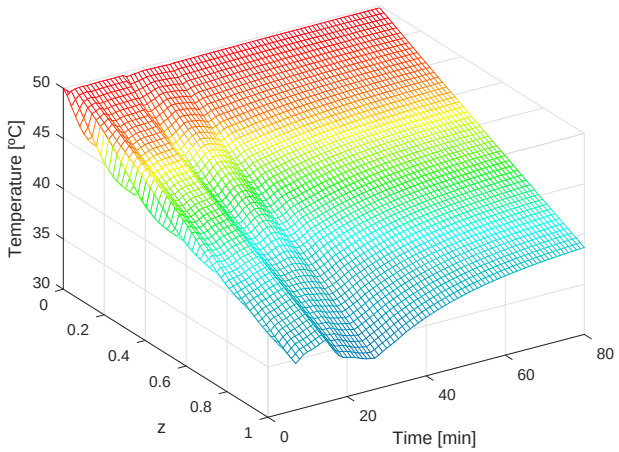

Figure 5.48: Simulation 9: Hot fluid temperature profile.

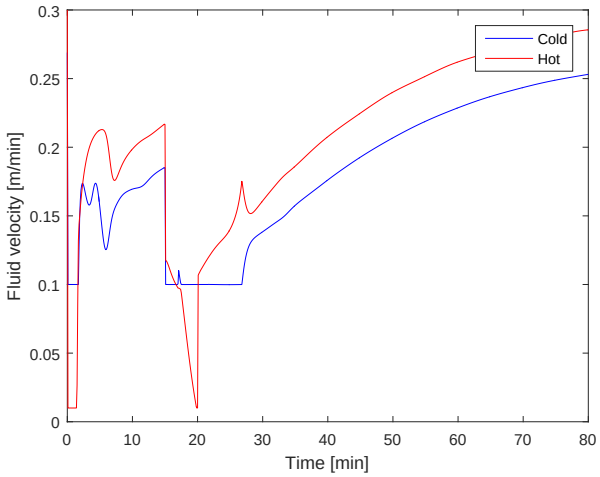

Figure 5.49: Simulation 9: Hot and cold fluid velocity [m/min].

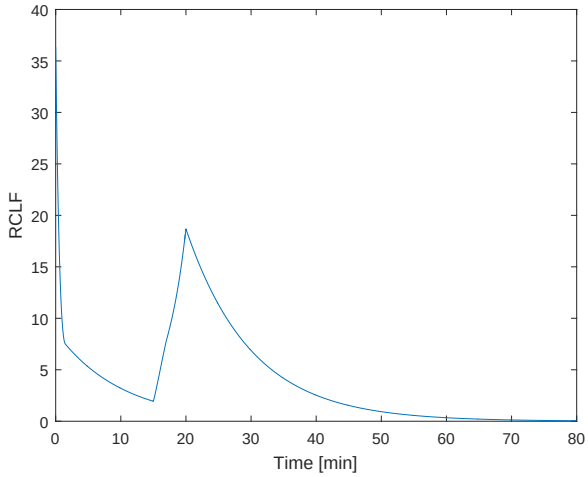

Figure 5.50: Simulation 9: Robust control Lyapunov function.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The study of novel control techniques applied to distributed parameter systems, namely hyperbolic nonlinear systems described by partial differential equations and involving parameter uncertainty were accomplished with success. These type of nonlinear systems were chosen, because they are who best describe transport phenomena processes in chemical and biological engineering, as well as other important engineering disciplines.

Model predictive control or receding horizon control of nonlinear systems has been proven to be a powerful tool on control of DPS and its combination with adaptation laws, in the sense of Lyapunov, and stability constraints were derived and applied successfully to tubular bioreactors and water delivery canals. A stable adaptive NMPC formulation for an uncertain class of distributed parameter biosystems was introduced and exemplified in chapter 3 in the form of a fixed-bed tubular bioreactor with Contois kinetics with the application of a stabilizing NMPC and LAL. Also, a general stability condition for the combined NMPC nominal formulation with a stable parameter observer was also successfully developed. Online parameter estimation, in order to deal with uncertainty, has been shown as well as the maintenance of low computational effort for closed-loop stability, making it a very decisive approach on solution of these kind of systems. The series of simulations performed demonstrated that the controller performance was maintained and that the inclusion of adaptation mechanisms applied in nonlinear hyperbolic DPS seems to be a natural extension on the NMPC.

The application of robust pointwise min-norm methedologies to a general result for distributed fluid flow systems stabilization around a stationary space profile was successfully obtained. This optimization statement, interpreted as a stabilizing limit solution when included in a predictive receding horizon control formulation results in closed-loop stability for any given horizon. The viability of alternative control techniques such as this has been proven, although costing in an increase on the computational effort.

Water distribution canals controlled by RPWMN control form a basis of more complex studies on canal engineered architectures resulting in the combination of robust and predictive design methods in order to achieve a fair compromise between water resources management and disturbance rejection. The extension from SISO case to MIMO cases were studied in both centralized and decentralized control architectures, respectively. For the particular MIMO case, control appears to be implemented in sequence, although after some time the optimization appear to be simultaneous when interactions increases. Computational effort in the decentralized control of the MIMO version appears to maintain a similar behavior exposed in the SISO centralized version.

There has also been studied the application of a locally distributed version of SIORHC, a form of model predictive controller for linear systems with constraints on the terminal state that ensures closed-loop stability. A set of multiple local agents that act in a neighbour-to-neighbour cooperative way to achieve a common goal - global optimal. The distributed control agents where projected to predict their control actions in a finite horizon relying in a local model that assumes only interaction with neighboring subsystems, via the input signals. The cooperation among these agents is accomplished when using an iterative procedure that converges to the optimal decentralized (*i.e.* fully-iterated) solution. Adjustments of weights in each subsystem cost function, ranging from a so-called egocentric behavior to a pure altruistic one resulted in different types of closed-loop performances and minimal requirements of a generic open computational infrastructure, where the cooperation of the different agents was shown when in presence of a sharing and broadcasting information service. In terms of closed-loop stability, results have shown that these agents can be directly applied to stable and unstable large scale linear systems. Finally, the application on the water delivery canal prototype earlier presented have produce encouraging results which allowed the validation of this new approach to distributed control.

Thorough dynamical coupled systems have also been studied in the form of counter-current heat exchangers. Again, adaptation combined with pointwise min-norm control have been proven to be an important control strategy for uncertain nonlinear systems described by partial differential equations. The application of this type of control on distributed and centralized control architectures shows that performance and robustness is maintained under difficult scenarios, corroborating and extending the results of chapter 3 for water delivery canals.

In terms of future work, the uncertain DPS systems and prototypes presented are being prepared for the extension to global distributed architectures in NMPC. By the same time as the writing of this thesis, a multivariable NMPC for the set of hyperbolic partial differential equations for the water delivery canals has been developed. Based upon some of the work here presented, a general formulation along with a computational efficient algorithm has been presented, using model reduction via the orthogonal collocation method resulting in two numerical examples that illustrates the application with success. Furthermore, decentralized and distributed architectures are being developed for the fixed-bed tubular

bioreactor presented in chapter 3 as a way to incorporate local and global optimization. The application of some of the proposed predictive techniques, namely pointwise min-norm control and receding horizon control with adaptation are under development for other hyperbolic and parabolic systems. Optimization of simulated moving bed liquid chromatography is already under development.

BIBLIOGRAPHY

[1] M. Abdelghani-Idrissi and F. Bagui. "Countercurrent double-pipe heat exchanger subjected to flow-rate step change, Part I: New Steady-State Formulation". In: *Heat Transfer Engineering* 23.5 (2002), pp. 4–11. DOI: 10.1080/01457630290090608.

[2] M. Abdelghani-Idrissi, F. Bagui, and L. Estel. "Countercurrent double-pipe heat exchanger subjected to flow-rate step change, Part II: Analytical and Experimental Transient Response". In: *Heat Transfer Engineering* 23.5 (2002), pp. 12–24. DOI: 10.1080/01457630290090617.

[3] V. Adetola, D. DeHaan, and M. Guay. "Adaptive Model Predictive Control for Constrained Nonlinear Systems." In: *Syst. Control Lett.* 58.5 (2009), pp. 320 –326. DOI: 10.1016/j.sysconle.2008.12.002.

[4] V. Adetola and M. Guay. "Adaptive Receding Horizon Control of Nonlinear Systems." In: *Proc. of the 6th IFAC-Symposium on Nonlinear Control Systems*. 2004.

[5] V. Adetola and M. Guay. "Robust Adaptive MPC for Constrained Uncertain Nonlinear Systems." In: *International Journal of Adaptive Control and Signal Processing* 25.2 (2011), pp. 155–167. DOI: 10.1002/acs.1193.

[6] A. Alessio and A. Bemporad. "Stability Conditions for Decentralized Model Predictive Control Under Packet Drop Communication." In: *Proc. of the American Control Conference 2008*. 2008, pp. 3577–3582. DOI: 10.1109/ACC.2008.4587048.

[7] B. Atkinson. *Bioprocess Engineering Principles.* Law Book Co. of Australasia, 1974. ISBN: 9780850860429.

[8] T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory.* Philadelphia: SIAM Society for Industrial and Applied Mathematics, 1999.

[9] M. Bakošová and J. Oravec. "Robust model predictive control for heat exchanger network". In: *Applied Thermal Engineering* 73.1 (2014), pp. 924 –930. ISSN: 1359-4311. DOI: 10.1016/j.applthermaleng.2014.08.023.

[10] L. Bakule. "Decentralized Control: An overview". In: *Annual Reviews in Control* 32.1 (2008), pp. 87 –98. DOI: 10.1016/j.arcontrol.2008.03.004.

[11] M. J. Balas. "The Galerkin Method and Feedback Control of Linear Distributed Parameter Systems." In: *Journal of Mathematical Analysis and Applications* 91.2 (1983), pp. 527 –546. DOI: 10.1016/0022-247X(83)90167-1.

[12] F. A. Barata, R. Campos, and R. Neves-Silva. "Distributed MPC for Thermal House Comfort with Shifting Loads and Limited Energy Resources." In: *Proc. of the 2013 International Conference on Renewable Energy Research and Applications (ICRERA)*. 2013, pp. 584–589. DOI: 10.1109/ICRERA.2013.6749823.

[13] F. A. Barata, N. Felix, and R. Neves-Silva. "Distributed MPC for Green Thermally Comfortable Buildings Based on an Electro-thermal Modular Approach." In: *Procedia Technology* 17.0 (2014). Conference on Electronics, Telecommunications and Computers CETC 2013., pp. 772 –780. DOI: 10.1016/j.protcy.2014.10.211.

[14] F. A. Barata, J. M. Igreja, and R. Neves-Silva. "Distributed MPC for Thermal Comfort and Load Allocation with Energy Auction." In: *Int. J. Renewable Energy Research* 4.2 (2014), pp. 371–383. ISSN: 1309-0127.

[15] F. A. Barata and R. Neves-Silva. "Distributed MPC for Thermal Comfort in Buildings with Dynamically Coupled Zones and Limited Energy Resources." In: *Technological Innovation for Collective Awareness Systems*. Ed. by L. Camarinha-Matos, N. Barrento, and R. Mendon. Vol. 423. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, 2014, pp. 305–312. ISBN: 978-3-642-54733-1. DOI: 10.1007/978-3-642-54734-8_34.

[16] G. Bastin and D. Dochain, eds. Process Measurement and Control. Amsterdam: Elsevier, 1990.

[17] A. Bemporad. "A Predictive Controller with Artificial Lyapunov Function for Linear Systems with Input/State Constraints." In: *Automatica* 34.10 (1998), pp. 1255 –1260. ISSN: 0005-1098. DOI: 10.1016/S0005-1098(98)00066-1.

[18] A. Bemporad and D. Barcelli. "Decentralized Model Predictive Control." In: *Networked Control Systems*. Vol. 406. Lecture Notes in Control and Information Sciences. Springer London, 2010, pp. 149–178.

[19] M. Berenguel and F. R. Rubio. *Advanced Control of Solar Plants.* Springer-Verlag London, 1997. ISBN: 978-1-4471-1249-5.

[20] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport Phenomena, Revised.* 2nd Ed. John Wiley & Sons, Inc., 2006. ISBN: 0470115394.

[21] D. M. Bošković and M. Krstić. "Backstepping Control of Chemical Tubular Reactors." In: *Comput. Chem. Eng.* 26.78 (2002), pp. 1077 –1085.

[22] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. "Distributed Model Predictive Control." In: *IEEE Control Systems* 22.1 (2002), pp. 44–52.

[23] Y. S. Cho and B. Joseph. "Reduced-order steady-state and dynamic models for separation processes. Part I. Development of the model reduction procedure". In: *AIChE Journal* 29.2 (1983), pp. 261–269. DOI: 10.1002/aic.690290213.

[24] P. D. Christofides. *Nonlinear and Robust Control of PDE Systems.* Birkhäuser Basel, 2001. ISBN: 978-0-8176-4156-6. DOI: 10.1007/978-1-4612-0185-4.

[25] S. J. Costa. "Técnicas Predictivas aplicadas a Controlo de Processos." MA thesis. Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Portugal, 2008.

[26] R. Courant and D. Hilbert. *Methods of Mathematical Physics: Volume II. Partial Differential Equations*. Wiley-VCH, 1989. ISBN: 978-0-471-50439-9.

[27] R. F. Curtain and H. J. Zwart. *An Introduction to Infinite-dimensional Linear Systems Theory.* Vol. 21. Texts in Applied Mathematics. Springer-Verlag, 1995. ISBN: 978-0-387-94475-3.

[28] L. Debnath. *Nonlinear Partial Differential Equations for Scientists and Engineers.* 3rd. Ed. Birkhäuser Boston, 2012. ISBN: 978-0-8176-8264-4.

[29] D. Dochain. "State Observation and Adaptive Linearizing Control for Distributed Parameter (Bio)chemical Reactors." In: *Int. J. Adapt. Control* 15.6 (2001), pp. 633–653. ISSN: 1099-1115. DOI: 10.1002/acs.691.

[30] D. Dochain, J. P. Babary, and N. Tali-Maamar. "Modelling and Adaptive Control of Nonlinear Distributed Parameter Bioreactors via Orthogonal Collocation." In: *Automatica* 28.5 (1992), pp. 873 –883. DOI: 10.1016/0005-1098(92)90141-2.

[31] P. M. Doran. *Bioprocess Engineering Principles.* Elsevier Science, 2012. ISBN: 978-0-12-220855-3.

[32] S. Dubljevic and P. D. Christofides. "Predictive control of parabolic PDEs with boundary control actuation." In: *Chem. Eng. Sci.* 61.18 (2006), pp. 6239–6248. ISSN: 0009-2509. DOI: 10.1016/j.ces.2006.05.041.

[33] S. Dubljevic, P. Mhaskar, N. H. El-Farra, and P. D. Christofides. "Predictive control of transport-reaction processes." In: *Comput. Chem. Eng.* 29.1112 (2005), pp. 2335 –2345. DOI: 10.1016/j.compchemeng.2005.05.008.

[34] F. Dubois, N. Petit, and P. Rouchon. "Motion Planning and Nonlinear Simulations for a Tank Containing a Fluid". In: *Proc. of the 1999 European Control Conference - ECC99*. 1999.

[35] M. Farina, G. Betti, and R. Scattolini. "Distributed predictive control of continuous-time systems." In: *Syst. Control Lett.* 74 (2014), pp. 32 –40. ISSN: 0167-6911. DOI: 10.1016/j.sysconle.2014.10.001.

[36] R. Findeisen and F. Allgöwer. "An Introduction to Nonlinear Model Predictive Control." In: *Proc. of the 21st Benelux Meeting on Systems and Control*. 2002.

[37] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. "Sur les Systs Nonlinres Diffntiellement Plats." In: *C.R. Acad. Sci. Paris* I (315 1992), pp. 619–624.

[38] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. "Flatness and Defect of Nonlinear Systems: Introductory theory and examples." In: *Int. J. Control* 61 (6 1995), pp. 1327–1361.

[39] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo. "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach." In: *IEEE Trans. Autom. Control* 53.1 (2008), pp. 324–338.

[40] R. Freeman and P. V. Kokotović. *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques.* 1st Ed. Birkhäuser, 1996. ISBN: 978-0817647582.

[41] R. A. Freeman and P. V. Kokotović. "Inverse Optimality in Robust Stabilization". In: *SIAM Journal of Control and Optimization* 34.4 (1996), pp. 1365–1391. ISSN: 0363-0129. DOI: 10.1137/S0363012993258732.

[42] P. Giselsson and A. Rantzer. "On Feasibility, Stability and Performance in Distributed Model Predictive Control". In: *Automatic Control, IEEE Transactions on* 59.4 (2014), pp. 1031–1036. ISSN: 0018-9286. DOI: 10.1109/TAC.2013.2285779.

[43] M. Gómez, J. Rodellar, and J. A. Mantecón. "Predictive Control Method for Decentralized Operation of Irrigation Canals." In: *Appl. Math. Model.* 26.11 (2002), pp. 1039 –1056. DOI: 10.1016/S0307-904X(02)00059-8.

[44] L. Grüne and J. Pannek. *Nonlinear and robust control of PDE systems.* Springer Verlag London, 2011. ISBN: 978-0-85729-500-2. DOI: 10.1007/978-0-85729-501-9.

[45] S. H., S. Jogwar, and P. Daoutidis. "Dynamics and control of high duty countercurrent heat exchangers". In: *Control Automation (MED), 2011 19th Mediterranean Conference on*. 2011, pp. 1034–1039. DOI: 10.1109/MED.2011.5983129.

[46] Y. He and J. Han. "Nonlinear Model Predictive Control Enhanced by Generalized Pointwise Min-Norm Scheme." In: *Proc. of the 46th IEEE Conference on Decision and Control*. 2007, pp. 4203–4208. DOI: 10.1109/CDC.2007.4434554.

[47] R. M. Hermans, M. Lazar, and A. Jokić. "Almost Decentralized Lyapunov-based Nonlinear Model Predictive Control." In: *Proc. of the American Control Conference*. 2010, pp. 3932–3938. DOI: 10.1109/ACC.2010.5530649.

[48] J. M. Igreja. "Mixed Constrained Stabilizing Predictive Control: An Extension to Multivariable Processes." In: *Proc. of the 5th Portuguese Conference on Automatic Control, Controlo 2002*. 2002, Aveiro, Portugal.

[49] J. M. Igreja and J. M. Lemos. "Nonlinear Model Predictive Control of a Water Distribution Canal Pool." English. In: *Nonlinear Model Predictive Control*. Vol. 384. Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, 2009, pp. 521–529. ISBN: 978-3-642-01093-4. DOI: 10.1007/978-3-642-01094-1_42.

[50] J. M. Igreja, J. M. Lemos, and S. Costa. "Controlo Geométrico de Sistemas Tubulares em Contra-Corrente". In: *Proc. of the 9th International Chemical Engineering Conference - CHEMPOR 2005*. 2005. ISBN: 972-8055-13-7.

[51] J. M. Igreja, J. M. Lemos, and S. J. Costa. "Adaptive Nonlinear Model Predictive Control for a Class of Transport Phenomena Processes." In: *Proc. of 8th IFAC Symposium on Nonlinear Control Systems, NOLCOS 2010*. 2010, pp. 729–734. ISBN: 978-3-902661-80-7. DOI: 10.3182/20100901-3-IT-2016.00100.

[52] J. M. Igreja, J. M. Lemos, and S. J. Costa. "Robust Pointwise Min-Norm Control of Distributed Systems with Fluid Flow." In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. 2011, pp. 2662–2667. DOI: 10.1109/CDC.2011.6160455.

[53] J. M. Igreja, J. M. Lemos, and R. N. Silva. "Adaptive Receding Horizon Control of Tubular Bioreactors." In: *Proc. of the 44th IEEE C.D.C. 2005 and 2005 E.C.C.* 2005, pp. 5168–5173. DOI: 10.1109/CDC.2005.1582982.

[54] J. M. Igreja, J. M. Lemos, and R. N. Silva. "Nonlinear Preditive Control of a Solar Plant Based on Reduced Complexity Models." In: *Proc. of the 16th IFAC World Congress*. 2005. DOI: 10.3182/20050703-6-CZ-1902.00787.

[55] J. M. Igreja, J. M. Lemos, and R. N. Silva. "Adaptive Control of Hyperbolic Systems: A CLF approach." In: *Proc. of the European Control Conference 2007*. 2007, pp. 2332–2339. ISBN: 978-960-89028-5-5.

[56] J. M. Igreja, J. M. Lemos, and R. N. Silva. "Controlling Distributed Hyperbolic Plants with Adaptive Nonlinear Model Predictive Control." In: *Assessment and Future Directions of Nonlinear Model Predictive Control*. Vol. 358. Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, 2007, pp. 435–441. ISBN: 978-3-540-72698-2. DOI: 10.1007/978-3-540-72699-9_35.

[57] J. M. Igreja, J. M. Lemos, P. Rouchon, and R. N. Silva. "Dynamic Motion Planning of a Distributed Collector Solar Field." In: *Proc. of the IFAC Symposium on Nonlinear Control Systems, NOLCOS 2004, Stuttgart, Germany*. 2004.

[58] J. M. Igreja, S. J. Costa, J. M. Lemos, and F. Cadete. "Computational Intelligence and Decision Making: Trends and Applications". In: Springer Netherlands, 2013. Chap. Multi-agent Predictive Control with Application in Intelligent Infrastructures, pp. 121–130. DOI: 10.1007/978-94-007-4722-7_12.

[59] A. Isidori. *Nonlinear Control Systems.* 3rd Ed. Springer, 1995. ISBN: 978-3540199168.

[60] T. Jahn and J. Pannek. "Stability of constrained adaptive Model Predictive Control algorithms." In: *proc. of the 18th IFAC World Congress*. Milan, Italy, 2011, pp. 9272–9277.

[61] I. Karafyllis and Z. Jiang. *Stability and Stabilization of Nonlinear Systems.* 1st Ed. Springer-Verlag London, 2011. ISBN: 978-0-85729-512-5.

[62] C. M. Kellett and P. M. Dower. "A Generalization of Input-to-State Stability." In: *IEEE 51st Annual Conference on Decision and Control (CDC2012)*. 2012, pp. 2970–2975. DOI: 10.1109/CDC.2012.6426008.

[63] B. Kestner. "Model Predictive Control (MPC) Algorithm for Tip-Jet Reaction Drive Systems." PhD thesis. Georgia Institute of Technology, USA, 2009.

[64] T. Keviczky, F. Borrelli, and G. J. Balas. "Distributed Predictive Control: Synthesis, Stability and Feasibility." In: *Cooperative Control of Distributed Multi-Agent Systems*. John Wiley & Sons, Ltd, 2007, pp. 79–108. DOI: 10.1002/9780470724200.ch5.

[65] A. Kharitonov and O. Sawodny. "Flatness-based feedforward control for parabolic distributed parameter systems with distributed control". In: *Int. J. Control* 79.7 (2006), pp. 677–687. DOI: 10.1080/00207170600622858.

[66] H. Krstić, I. Kanellakopoulos, and P. Kokotović. *Nonlinear and Adaptive Control Design.* 1st Ed. Wiley-Interscience, 1995. ISBN: 978-0471127321.

[67] J. La Salle and S. Lefschetz. *Stability by Liapunov's Direct Methods with Applications.* 1st Ed. Academic Press, 1961.

[68] L. Lefèvre, D. Dochain, S. F. de Azevedo, and A. Magnus. "Optimal Selection of Orthogonal Polynomials applied to the Integration of Chemical Reactor Equations by Collocation Methods." In: *Comput. Chem. Eng.* 24.12 (2000), pp. 2571 –2588. ISSN: 0098-1354. DOI: 10.1016/S0098-1354(00)00597-4.

[69] J. M. Lemos and J. M. Igreja. "D-SIORHC, a Distributed MPC with Stability Constraints Based on a Game Approach." In: *Distributed MPC Made Easy*. Springer, 2014.

[70] J. M. Lemos, L. M. Rato, F. Machado, N. Nogueira, P. Salgueiro, R. N. Silva, and M. Rijo. "Predictive Adaptive Control of Water Level in Canal Pools." In: *Proc. of the 16th Int. Conf. Systems Science, Wroclaw, Poland*. 2007, pp. 139–148.

[71] O. Levenspiel. *Chemical Reaction Engineering.* 3rd Ed. John Wiley & Sons, 1998. ISBN: 978-0471254249.

[72] J. Lévine. *Analysis and Control of Nonlinear Systems: A Flatness-based Approach.* 1st Ed. Springer, 2009. ISBN: 978-3-642-00838-2.

[73] D. Limon, T. Alamo, D. M. Raimondo, D. M. de la Peña, J. M. Bravo, A. Ferramosca, and E. F. Camacho. "Input-to-State Stability: A Unifying Framework for Robust Model Predictive Control." In: *Nonlinear Model Predictive Control*. Ed. by L. Magni, D. M. Raimondo, and F. Allgöwer. Vol. 384. Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, 2009, pp. 1–26. ISBN: 978-3-642-01093-4. DOI: 10.1007/978-3-642-01094-1_1.

[74] X. Litrico and V. Fromion. "H^∞ control of an Irrigation Canal Pool with a Mixed Control Politics." In: *IEEE Transactions on Control Systems Technology* 14.1 (2006), pp. 99–111. DOI: 10.1109/TCST.2005.860526.

[75] J. Liu, X. Chen, D. M. de la Peña, and P. D. Christofides. "Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. Part I: Theory". In: *Proc. of American Control Conference*. 2010, pp. 3148–3155.

[76] J. Liu, X. Chen, D. M. de la Peña, and P. D. Christofides. "Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. Part II: Application to a catalytic alkylation of benzene process." In: *Proc. of American Control Conference*. 2010, pp. 3156–3161.

[77] J. Liu, X. Chen, D. D. M. de la Peña, and P. D. Christofides. "Iterative Distributed Model Predictive Control of Nonlinear Systems: Handling Asynchronous, Delayed Measurements". In: *IEEE Trans. Autom. Control* 57.2 (2012), pp. 528–534. ISSN: 0018-9286. DOI: 10.1109/TAC.2011.2164729.

[78] J. M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, 2002.

[79] J. M. Maestre, D. D. M. de la Peña, and E. F. Camacho. "Distributed model predictive control based on a cooperative game". In: *Optimal Control Applications and Methods* 32.2 (2011), pp. 153–176. ISSN: 1099-1514. DOI: 10.1002/oca.940.

[80] L. Magni and R. Scattolini. "Stabilizing Decentralized Model Predictive Control of Nonlinear Systems." In: *Automatica* 42.7 (2006), pp. 1231–1236. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2006.02.010.

[81] L. Magni, G. Nicolao, L. Magnani, and R. Scattolini. "A Stabilizing Model-based Predictive Control Algorithm for Nonlinear Systems." In: *Automatica* 37.9 (2001), pp. 1351–1362. DOI: 10.1016/S0005-1098(01)00083-8.

[82] A. Maidi, M. Diaf, and J. Corriou. "Boundary geometric control of a counter-current heat exchanger". In: *Journal of Process Control* 19.2 (2009), pp. 297 –313. ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2008.03.002.

[83] W. Marquardt. "Traveling Waves in Chemical Processes." In: *Int. Chem. Eng.* 30.4 (1990), pp. 585–606.

[84] P. Martin, R. M. Murray, and P. Rouchon. "Flat Systems." In: *Lecture Notes given at the Summer School on Mathematical Control Theory, Trieste, Italy*. 2001.

[85] D. Q. Mayne. "Model predictive control: Recent developments and future promise". In: *Automatica* 50.12 (2014), pp. 2967 –2986.

[86] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. "Constrained Model Predictive Control: Stability and optimality." In: *Automatica* 36.6 (2000), pp. 789 –814. DOI: 10.1016/S0005-1098(99)00214-9.

[87] M. Mercangöz and F. J. Doyle. "Distributed Model Predictive Control of a Four-Tank System." In: *Proc. of the International Symposium on Advanced Control of Chemical Processes, ADCHEM2006, Gramado, Brazil*. 2006, pp. 965–970.

[88] E. Mosca. *Optimal, Predictive and Adaptive Control.* 1st Ed. Prentice Hall, 1995. ISBN: 978-0138476090.

[89] E. Mosca, J. M. Lemos, and J. Zhang. "Stabilizing I/O Receding Horizon Control." In: *Proc. of the 29th IEEE Conference on Decision and Control, 1990.* Vol. 4. 1990, 2518–2523 vol.4. DOI: 10.1109/CDC.1990.203454.

[90] M. A. Müller, M. Reble, and F. Allgöwer. "Cooperative Control of Dynamically Decoupled Systems via Distributed Model Predictive Control." In: *Int. J. Robust Nonlin.* 22.12 (2012), pp. 1376–1397. DOI: 10.1002/rnc.2826.

[91] R. R. Negenborn. "Multi-agent Model Predictive Control with Applications to Power Networks." PhD thesis. Technische Universiteit Delft, Utrecht, 2007.

[92] M. T. Nihtila, J. Tervo, J. P. Kaipio, and J. P. Babary. "Controller Design Issues and Algorithms for a Nonlinear Distributed-parameter Process." In: *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*. Vol. 3. 1997, pp. 2414–2419. DOI: 10.1109/ICSMC.1997.635289.

[93] S. Nithya, A. Gour, N. Sivakumaran, T. Radhakrishnan, and N. Anantharaman. "Predictive controller design for a shell and tube heat exchanger". In: *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*. 2007, pp. 1075–1080. DOI: 10.1109/ICIAS.2007.4658550.

[94] A. Osório, H. Afsarmanesh, and L. Camarinha-Matos. "Open Services Ecosystem Supporting Collaborative Networks." In: *Balanced Automation Systems for Future Manufacturing Networks*. Vol. 322. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, 2010, pp. 80–91. ISBN: 978-3-642-14340-3. DOI: 10.1007/978-3-642-14341-0_10.

[95] J. Primbs, V. Nevistić, and J. C. Doyle. "A Receding Horizon Generalization of Pointwise Min-Norm Controllers." In: *IEEE T. Automat. Contr.* 45 (2000), pp. 898–909.

[96] J. A. Primbs, V. Nevistić, and J. C. Doyle. "Nonlinear Optimal Control: A Control Lyapunov Function and Receding Horizon Perspective." In: *Asian Journal of Control* 1.1 (1999), pp. 14–24. DOI: 10.1111/j.1934-6093.1999.tb00002.x.

[97] S. J. Qin and T. A. Badgwell. "A Survey of Industrial Model Predictive Control Technology." In: *Control Eng. Pract.* 11.7 (2003), pp. 733 –764. ISSN: 0967-0661. DOI: 10.1016/S0967-0661(02)00186-7.

[98] D. M. Raimondo, L. Magni, and R. Scattolini. "Decentralized MPC of Nonlinear Systems: An input-to-state stability approach." In: *Int. J. Robust Nonlin.* 17.17 (2007), pp. 1651–1667. DOI: 10.1002/rnc.1214.

[99] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control Theory and Design.* Nob Hill Pub., 2009. ISBN: 978-0975937709.

[100] J. Rawlings. "Tutorial overview of model predictive control". In: *Control Systems, IEEE* 20.3 (2000), pp. 38–52. ISSN: 1066-033X. DOI: 10.1109/37.845037.

[101] R. G. Rice and D. D. Do. *Applied Mathematics and Modeling for Chemical Engineers.* 1st Ed. John Wiley & Sons, 1995. ISBN: 978-0471303770.

[102] P. Rouchon. "Motion Planning, Equivalence, Infinite Dimensional Systems." In: *Int. J. Appl. Math. Comput. Sci.* 11 (2001), pp. 165–188.

[103] J. Rudolph. *Flatness Based Control of Distributed Parameter Systems.* Shaker Verlag, 2003.

[104] W. J. Rugh. *Linear System Theory.* Prentice Hall, 1996. ISBN: 9780134412054.

[105] R. Scattolini. "Architectures for Distributed and Hierarchical Model Predictive Control A review." In: *J. Process Contr.* 19.5 (2009), pp. 723 –731.

[106] H. Shang, J. F. Forbes, and M. Guay. "Model Predictive Control for Quasilinear Hyperbolic Distributed Parameter Systems." In: *Ind. Eng. Chem. Res.* 43.9 (2004), pp. 2140–2149. DOI: 10.1021/ie030653z.

[107] P. B. Sistu and B. W. Bequette. "Nonlinear model-predictive control: Closed-loop stability analysis". In: *AIChE Journal* 42.12 (1996), pp. 3388–3402. ISSN: 1547-5905. DOI: 10.1002/aic.690421210.

[108] S. Skogestad. "Control Structure Design for Complete Chemical Plants." In: *Computers & Chemical Engineering* 28.12 (2004). Escape 12, pp. 219 –234. ISSN: 0098-1354. DOI: 10.1016/j.compchemeng.2003.08.002.

[109] E. D. Sontag. "A 'universal' construction of Artstein's theorem on nonlinear stabilization." In: *Systems & Control Letters* 13.2 (1989), pp. 117 –123. ISSN: 0167-6911. DOI: 10.1016/0167-6911(89)90028-5.

[110] E. D. Sontag. "Smooth stabilization implies coprime factorization." In: *IEEE Trans. Autom. Control* 34.4 (1989), pp. 435–443. DOI: 10.1109/9.28018.

[111] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems.* 2nd Ed. New York, NY, USA: Springer-Verlag New York, Inc., 1998. ISBN: 978-1-4612-6825-3. DOI: 10.1007/978-1-4612-0577-7.

[112] E. D. Sontag. "Input to State Stability: Basic Concepts and Results." In: *Nonlinear and Optimal Control Theory*. Ed. by P. Nistri and G. Stefani. Vol. 1932. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 2008, pp. 163–220. ISBN: 978-3-540-77644-4. DOI: 10.1007/978-3-540-77653-6_3.

[113] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. "Cooperative Distributed Model Predictive Control." In: *Syst. Control Lett.* 59.8 (2010), pp. 460 –469. DOI: 10.1016/j.sysconle.2010.06.005.

[114] A. Teel, R. Kadiyal, P. Kokotović, and S. Sastry. "Indirect Techniques for Adaptive Input-Output Linearization of Non-linear Systems." In: *Int. J. Control* 53.1 (1991), pp. 193–222. DOI: 10.1080/00207179108953617.

[115] M. J. Tippett and J. Bao. "Distributed Control of Chemical Process Networks." In: *Int. J. of Automation and Computing* 12.4 (2015), pp. 368–381. ISSN: 1476-8186. DOI: 10.1007/s11633-015-0895-9.

[116] M. Vaccarini, S. Longhi, and R. Katebi. "State Space Stability Analysis of Unconstrained Decentralized Model Predictive Control Systems." In: *American Control Conference*. 2006, 6 pp.–. DOI: 10.1109/ACC.2006.1655347.

[117] A. N. Venkat, J. B. Rawlings, and S. J. Wright. "Plant-wide Optimal Control with Decentralized MPC." In: *Proc. of the 7th International Symposium on Dynamics and Control of Process Systems (DYCOPS 7)*. 2004.

[118] A. N. Venkat, J. B. Rawlings, and S. J. Wright. "Stability and Optimality of Distributed Model Predictive Control." In: *46th IEEE Conference on Decision and Control (CDC2005)*. 2005, pp. 6680–6685. DOI: 10.1109/CDC.2005.1583235.

[119] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. "Distributed MPC Strategies with Application to Power System Automatic Generation Control." In: *IEEE Trans. Control Syst. Technol.* 16.6 (2008), pp. 1192–1206. ISSN: 1063-6536. DOI: 10.1109/TCST.2008.919414.

[120] J. Villadsen and W. Stewart. "Solution of boundary-value problems by orthogonal collocation". In: *Chemical Engineering Science* 22.11 (1967), pp. 1483–1501. ISSN: 0009-2509. DOI: 10.1016/0009-2509(67)80074-5.

[121] D. D. Šiljak. *Decentralized Control of Complex Systems*. Dover Publications, 2012. ISBN: 978-0486486147.

[122] D. D. Šiljak and A. I. Zečević. "Control of Large-Scale Systems: Beyond decentralized feedback." In: *Annu. Rev. Control* 29.2 (2005), pp. 169 –179. ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2005.08.003.

[123] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara. "Experimental Analysis of Model Predictive Control for an Energy Efficient Building Heating System." In: *Appl. Energ.* 88.9 (2011), pp. 3079 –3087. DOI: 10.1016/j.apenergy.2011.03.009.

[124] S. Wang and E. Davison. "On the Stabilization of Decentralized Control Systems." In: *IEEE Trans. Autom. Control* 18.5 (1973), pp. 473–478. ISSN: 0018-9286.

[125] W. Wang, D. Rivera, and K. Kempf. "Centralized Model Predictive Control Strategies for Inventory Management in Semiconductor Manufacturing Supply Chains." In: *Proc. of the American Control Conference*. Vol. 1. 2003, pp. 585–590.

[126] Y. Wang and S. Boyd. "Performance bounds for linear stochastic control." In: *Syst. Control Lett.* 58.3 (2009), pp. 178–182.

[127] Y. Zhang and S. Li. "Networked Model Predictive Control based on Neighbourhood Optimization for Serially Connected Large-scale Processes." In: *J. Process Contr.* 17.1 (2007), pp. 37 –50. DOI: 10.1016/j.jprocont.2006.08.009.

APPENDIX A

APPENDIX

A.1 Pseudocode Algorithms

All routines were developed using *MathWorks® MATLAB*™ and converted here into pseudocode algorithms. The appendix is divided in three different sections, depending on the system in study.

A.2 Fixed-Bed Bioreactor routines

In chapter 2, six routines were developed, namely:

- RHCBioreactor as the main routine for predictive and adaptive control of the bioreactor system;
- Custobioreactormoc used to calculate the cost function for the reduced model;
- Colloc used to calculate OC points and differential matrices;
- Jacobi used to calculates Jacobi polynomial coefficients to be used in colloc routine;
- Dlagrange used to calculate OC matrices based on Lagrange polynomials and its derivatives;
- BioreactorAdapt used to calculate the output for a Bioreactor using an adaptive RH controller.

Algorithm 1 RHCBioreactor routine: Calculates the output for a Bioreactor using an adaptive RH controller

1: **Inputs**: u
2: **Begin**
3: Initializes Uk,$Sock$,$Xock$,$yref$,$Theta1k$,$Theta2k$,$Socpk$,$Xocpk$,Vks
4: Define the model parameters L,k_1,k_c,k_d,μ_b,A_r,α ▷ α is an auxiliary parameter
5: Define the initial inlet concentration S_{in} and initial velocity u_{ee}
6: $N_{int} \leftarrow \{value\}$ ▷ number of interior points
7: Uses subroutine colloc to calculate xoc,$Aoc1$ and $Boc1$ using N_{int};
8: $Aoc \leftarrow -Aoc1(2 : N_{int} + 2, 2 : N_{int} + 2)$
9: $Boc \leftarrow -Aoc1(2 : N_{int} + 2, 1)$
10: $Sooc \leftarrow e^{(-k_1 * (\mu_b - k_d)/uee/kc * xoc(2:N_{int}+2))} * S_{in}$ ▷ Calculates initial substrate in OCM
11: $Xooc \leftarrow (\mu_b - k_d)/k_c/k_d * Sooc$ ▷ Calculates initial biomass in OCM
12: $yooc \leftarrow [Sooc\ Xooc]'$ ▷ Puts both substrate and biomass in a output matrix form
13: $k_1est \leftarrow \{value\}$
14: $k_dest \leftarrow \{value\}$
15: $uk \leftarrow \{value\}$ ▷ Defines initial value of the RHC
16: $Vk \leftarrow \{value\}$ ▷ Defines initial value of the CLF
17: $Uo \leftarrow [uk \cdots uk]$ ▷ Builds the matrix for the prediction horizon
18: $I \leftarrow length(Uo)$
19: Define $tmax$, dt, $tsim$ and $kmax$ ▷ Calculates integration time, intervals and number of simulations
20: Define y_{ref} trajectory for substrate S

for $k = 1 \rightarrow kmax$ **do** ▷ Calculates states X^* and S^*
  $uee \leftarrow -k_1 * (\mu_b - k_d)/k_c * L/log(yref(k)/S_{in})$;
  $Sooc \leftarrow e^{(-k1*(muib-kd)/uee/kc*xoc(2:Nint+2))} * S_{in}$;
  if $k == 1$ **then**
    $Sk \leftarrow Sooc$;
  end if;
  $Xooc \leftarrow (\mu_b - k_d)/k_c/k_d * Sooc$;
  if $k == 1$ **then**
    $Xk \leftarrow Xooc$;
  end if;
  $Vk_0 \leftarrow (S_k - Sooc) * (S_k - Sooc)' + value * (X_k - Xooc) * (X_k - Xooc)'$; ▷ Calculates the CLF based on states S and X
  $alfa = k_{1est}/k_c * (\mu_b - k_{dest}) * L$;
  $uss \leftarrow -alfa * 1/log(y_{ref}(k)/S_{in})$;
  $S_{in} \leftarrow value$ ▷ Indicates inlet concentration of substrate
  Calculates the velocity matrix while minimizes the cost functional based on subroutine custobioreactmoc1 ▷ Minimization makes use of fmincon function in Matlab
  $u_k \leftarrow U(1)$;
  $Uo \leftarrow [Uo(2:I)U(I)]$;
  $u_k \leftarrow max(min(uk, uppervalue), lowervalue)$; ▷ Includes saturation on velocity u at each iteration
  Calculates the output based on adaptation routine BioreactorAdapt
  $yinis \leftarrow yocs(size(yocs,1),:)'$;
  $yooc \leftarrow yocs(size(yocs,1),1:2*Noc)'$;
  $Sk \leftarrow yocs(size(yocs,1),1:Noc)$; ▷ Stores substrate values calculated at iteration k
  $Xk \leftarrow yocs(size(yocs,1),Noc+1:2*Noc)$; ▷ Stores biomass values calculated at iteration k
  $Vk \leftarrow (Sk - Sooc) * (Sk - Sooc)' + value * (Xk - Xooc) * (Xk - Xooc)'$;
  $Sock \leftarrow [Sock\ ;\ yocs(size(yocs,1),1:Noc)]$;
  $Xock \leftarrow [Xock\ ;\ yocs(size(yocs,1),Noc+1:2*Noc)]$;
  $Socpk \leftarrow [Socpk\ ;\ yocs(size(yocs,1),2*Noc+1:3*Noc)]$;
  $Xocpk \leftarrow [Xocpk\ ;\ yocs(size(yocs,1),3*Noc+1:4*Noc)]$;
  $Theta1k = [Theta1k\ ;\ yocs(size(yocs,1),4*Noc+1)]$;
  $Theta2k = [Theta2k\ ;\ yocs(size(yocs,1),4*Noc+2)]$;
  $y \leftarrow yocs(size(yocs,1),Noc)$;
  $k1est \leftarrow yocs(size(yocs,1),4*Noc+1)$; ▷ Estimated k_1 through adaptation
  $kdest \leftarrow yocs(size(yocs,1),4*Noc+2)$; ▷ Estimated k_d through adaptation
  $Uk \leftarrow [Uk\ uk]$;
  $Vks = [Vks\ Vk]$;
end for

Algorithm 2 Custobioreactormoc routine: Calculates cost function for the system using Orthogonal collocation method

Inputs: u,dt,$xini$,Aoc,Boc,k_1,k_c,k_d,μ_b,S_{in},$xref$,Noc,uss
Outputs: $Jrhc$
Begin
$I = length(u)$;
Initializes Xs and ts ▷ Xs is states and ts is time
for $i = 1 : I$ **do**
    Integrates Bioreactor model based on initial states, inputs and parameters
    $xini = x(size(x,1),:)'$;
    $Xs = [Xs\ ;\ x(:,1:Noc)]$;
    $ts = [ts\ ;\ t]$;
end for;
Starts the calculations on cost integral using trapezoid method
$lt = length(ts)$;
$Dus \leftarrow u - [uss\ u(1:I-1)]$;
$Dts \leftarrow ts(2:lt) - ts(1:lt-1)$;
$S1 \leftarrow (Xs(1:lt-1, size(Xs,2)) + Xs(2:lt, size(Xs,2)))/2$;
$S3 \leftarrow sum(Dus.^2) * dt$;
$Jrhc \leftarrow value * ((S1 - xref)'.^2 * Dts + weoght * S3)$; ▷ weight refers to control action weight

Algorithm 3 Colloc routine: Calculates Orthogonal Collocation points and differential matrices accordingly to [101]

Inputs: α,β,N
Outputs: γ
Begin
$\gamma \leftarrow jacobi(\alpha, \beta, N)$; ▷ Calculates Jacobi polynomial coefficients based on Jacobi routine
$x \leftarrow [0\ fliplr((roots(\gamma))')\ 1]$; ▷ Calculates the zeros for Jacobi polynomial
$[A, B] \leftarrow dlagrange(x)$; ▷ Uses dlagrange routine to calculate first and second order derivatives for Lagrange polynomial

Algorithm 4 Jacobi routine: Calculates Jacobi polynomial coefficients

Inputs: α,β,N
Outputs: γ
Begin
$\gamma(1) \leftarrow 1$;
for $i = 1 : N$ **do**
    $\gamma(i+1) \leftarrow (N - i + 1) * (N + i + \alpha + \beta) * \gamma(i)/i/(i + \beta)$;
end for
$\gamma \leftarrow fliplr(\gamma. * ((-1).^(N - (0:N))))$;

Algorithm 5 Dlagrange routine: Calculates Orthogonal Collocation matrices based on Lagrange polynomials and its derivatives

Inputs: x
Outputs: A and B matrices for OC containing Lagrange polynomial first and second order derivatives accordingly to [101].
Begin
$M \leftarrow length(x)$; ▷ M is $N+2$ interior collocation points
Initializes $P1$,$P2$,$P3$;
for $i = 1 : M$ **do**
    $p \leftarrow 1$;,$p1 \leftarrow 0$;,$p2 \leftarrow 0$;,$p3 \leftarrow 0$;
    for $j = 1 : M$ **do**
        $dx \leftarrow (x(i) - x(j))$;
        $p3 \leftarrow dx * p3 + 3 * p2$;
        $p2 \leftarrow dx * p2 + 2 * p1$;
        $p1 \leftarrow dx * p1 + p$;
        $p \leftarrow dx * p$;
    end for
    $P1 \leftarrow [P1\ p1]$;
    $P2 \leftarrow [P2\ p2]$;
    $P3 \leftarrow [P3\ p3]$;
end for
for $i = 1 : M$ **do**
    for $j = 1 : M$ **do**
        if $i == j$ **then**
            $A(i,j) = 1/2 * P2(i)/P1(i)$;
        else$A(i,j) = 1/(x(i) - x(j)) * P1(i)/P1(j)$;
        end if
    end for;
end for;
for $i = 1 : M$ **do**
    for $j = 1 : M$ **do**
        if $i == j$ **then**
            $B(i,j) = 1/3 * P3(i)/P1(i)$;
        else$B(i,j) = 2 * A(i,j) * (A(i,i) - 1/(x(i) - x(j)))$;
        end if
    end for;
end for;

Algorithm 6 BioreactorAdapt routine: Calculates the output for a Bioreactor using an adaptive RH controller

Inputs: t,y,A,B,k_1,k_c,k_d,μ_b,u,so,N
Begin
$s \leftarrow y(1:N)$;
$x \leftarrow y(N+1:2*N)$;
$st \leftarrow y(2*N+1:3*N)$;
$xt \leftarrow y(3*N+1:4*N)$;
$theta1 \leftarrow y(4*N+1)$;
$theta2 \leftarrow y(4*N+2)$;
$\mu \leftarrow (1./(k_c*x+s)).*s*\mu_b$; ▷ Element-wise operations uses dot
$sp \leftarrow u*(A*s+B*so) - k_1*\mu.*x$;
$xp \leftarrow (\mu - k_d).*x$;
$stp \leftarrow u*(A*s+B*so) - theta1*\mu.*x + value*ones(1,N)*(s-st)$;
$xtp \leftarrow (\mu - theta2).*x + value*ones(1,N)*(x-xt)$;
$t1p \leftarrow -value1*(\mu.*x)'*(s-st)$; ▷ $value1$ is the adaptive controller weight for substrate
$t2p \leftarrow -value2*x'*(x-xt)$; ▷ $value2$ is the adaptive controller weight for biomass

A.3 Water Delivery Canal Pools

In chapter 3, two main routines, two model routines and a constraint routine were developed, namely:

- Canal_onepool as the main routine for PWMN control in a single water canal pool;
- Canal_twopools as the main routine for PWMN control in two water canal pools;
- Canalmod100 used to define the model for one water canal pool (includes optimization);
- Canalmod200 used to define the model for two water canal pools (includes optimization for centralized and decentralized control);
- SC used to include the constraints of the optimization problem (this function is included in both model algorithms).

Algorithm 7 Canal_onepool routine: Calculates the output for a Water delivery canal pool using a PWMN controller.

Inputs: Initial states and inputs
Outputs: matrix Y containing water elevation h, water velocity v and gate opening u
Begin
Define physical parameters:
g ▷ gravitational constant
$cd12$ ▷ Discharge coefficient
n ▷ Manning coefficient
b ▷ Bottom width
d ▷ Trapezoid slope
J ▷ Canal slope
L ▷ Pool length
A_d ▷ Discharge area
nl ▷ Number of pool sections
$dx \leftarrow L/nl$ ▷ Length of pool section
Auxiliary calculations and matrices:
n^2
$sd \leftarrow \sqrt{1+d^2}$
$cc \leftarrow cd12 * A_d$
$A3 \leftarrow [zeros(nl-2,1)\ eye(nl-2); zeros(1,nl-1)] - [[zeros(1,nl-2); eye(nl-2)] zeros(nl-1,1)]$
$A2 = [[2\ zeros(1,nl-2)]; A3; [zeros(1,nl-2)\ -2]]$
$A1 = [[-2\ ;\ -1\ ;\ zeros(nl-1,1)]\ A2\ [zeros(nl-1,1)\ ;\ 1;\ 2]];$
Recalculates matrix $A1$ as $A1 \leftarrow 0.5 * A1$
Recalculates matrix $A2$ as $A2 \leftarrow A1(2:nl,:)$
Defines initial conditions for the simulation
$x \leftarrow 0 : dx : L$
Defines initial water elevation $h \leftarrow value * ones(nl+1,1)$
Defines initial water velocity $v \leftarrow value * ones(nl-1,1)$
Defines initial upstream gate position $u1$
Defines initial downstream gate position $u2$
Defines upstream elevation H_u
Defines upstream elevation H_d
Integrates the model in $canalmod100$ to obtain water elevation h, water velocity v and gate opening u

Algorithm 8 Canal_twopools routine: Calculates the output for a Water delivery canal pools using a centralized or decentralized PWMN controllers.

Inputs: Initial states and inputs
Outputs: matrix Y containing water elevations $h1$ and $h2$, water velocities $v1$ and $v2$ and gates opening u_i $(i = 1, 2, 3)$
Begin
Define physical parameters:
g ▷ gravitational constant
$cd12$ ▷ Discharge coefficient
n ▷ Manning coefficient
b ▷ Bottom width
d ▷ Trapezoid slope
J ▷ Canal slope
L ▷ Pool length
A_d ▷ Discharge area
nl ▷ Number of pool sections
$dx \leftarrow L/nl$ ▷ Length of pool section
Auxiliary calculations and matrices:
n^2
$sd \leftarrow \sqrt{1 + d^2}$
$cc \leftarrow cd12 * A_d$
$A3 \leftarrow [zeros(nl-2,1)eye(nl-2); zeros(1,nl-1)] - [[zeros(1,nl-2); eye(nl-2)]zeros(nl-1,1)]$
$A2 = [[2\ zeros(1,nl-2)]; A3; [zeros(1,nl-2)\ -2]]$
$A1 = [[-2\ ;\ -1; zeros(nl-1,1)]\ A2\ [zeros(nl-1,1)\ ;\ 1;\ 2]];$
Recalculates matrix $A1$ as $A1 \leftarrow 0.5 * A1$
Recalculates matrix $A2$ as $A2 \leftarrow A1(2 : nl, :)$
Defines initial conditions for the simulation
$x \leftarrow 0 : dx : L$
Defines initial water elevation for first pool $h1 \leftarrow value * ones(nl + 1, 1)$
Defines initial water elevation for second pool $h2 \leftarrow value * ones(nl + 1, 1)$
Defines initial water velocity for first pool $v1 \leftarrow value * ones(nl - 1, 1)$
Defines initial water velocity for second pool $v2 \leftarrow value * ones(nl - 1, 1)$
Defines initial upstream gate position $u1$
Defines initial middle gate position $u2$
Defines initial downstream gate position $u3$
Defines upstream elevation Hu
Defines upstream elevation Hd
Integrates the model in canalmod200 to obtain water elevations $h1$ and $h2$, water velocities $v1$ and $v2$ and gates opening u_i, $(i = 1, 2, 3)$ depending on which gates are being controlled.

Algorithm 9 Canalmod100 routine: Model for a Water delivery canal pool.

Inputs: Initial states, inputs and parameters
Outputs: matrix Y containing water elevations $h1$ and $h2$, water velocities $v1$ and $v2$ and gates opening u_i $(i = 1, 2, 3)$
Begin
Positions the water elevation in the output matrix $h \leftarrow y(1 : nl + 1)$
Positions the water velocity in the output matrix $v \leftarrow y(nl + 2 : 2 * nl)$
$ah \leftarrow b * h + d * h. * h$
$fh \leftarrow ah./(b + 2 * d * h)$
$rh \leftarrow n2./(ah/b + 2 * h * sd).^{(}4/3)$
$dh1 \leftarrow H1 - h(1)$
$dh2 \leftarrow h(nl + 1) - H2$
Runs optimization routine using MATLAB fmincon function subject to the SC routine
$u1 \leftarrow y(end)$
$u1 \leftarrow max(min(u1, 0.5), 0)$ ▷ Saturates gate opening
$v0 \leftarrow sign(dh1) * cc * u1 * sqrt(2 * g * abs(dh1))/ah(1)$
$vN \leftarrow sign(dh2) * cc * u2 * sqrt(2 * g * abs(dh2))/ah(nl + 1)$
$vbc \leftarrow [v0\ ;\ v\ ;\ vN]$ ▷ Boundary velocities vector
$hp \leftarrow -(vbc. * (A1 * h) + fh. * (A1 * vbc))/dx$
$vp \leftarrow -(g * A2 * h + v. * (A2 * vbc))/dx - g * (v. * abs(v). * rh(2 : nl) - J);$
$up \leftarrow (-y(end) + u1) * value$ ▷ Filtered input
$yp \leftarrow [hp\ ;\ vp;\ up];$ ▷ Output matrix

Algorithm 10 Canalmod200 routine: Model for two water delivery canal pools in series.

Inputs: Initial states, inputs and parameters
Outputs: matrix Y containing water elevations $h1$ and $h2$, water velocities $v1$ and $v2$ and gates opening u_i $(i = 1, 2, 3)$
Begin
Positions the water elevation of the first pool in the output matrix $h1 \leftarrow y(1 : nl + 1)$
Positions the water velocity of the first pool in the output matrix $v1 \leftarrow y(nl + 2 : 2 * nl)$
Positions the water elevation of the second pool in the output matrix $h2 \leftarrow y(2 * nl + 1 : 3 * nl + 1)$
Positions the water velocity of the second pool in the output matrix $v2 \leftarrow y(3 * nl + 2 : 4 * nl)$
$ah1 \leftarrow b * h1 + d * h1. * h1$
$ah2 \leftarrow b * h2 + d * h2. * h2$
$fh1 \leftarrow ah1./(b + 2 * d * h1$
$fh2 \leftarrow ah2./(b + 2 * d * h2)$
$rh1 \leftarrow n^2./(ah1/b + 2 * h1 * sd).^(4/3)$
$rh2 \leftarrow n^2./(ah2/b + 2 * h2 * sd).^(4/3)$
$dh1 \leftarrow H1 - h(1)$
$dh2 \leftarrow h1(nl + 1) - h2(1)$
$dh3 \leftarrow h2(nl + 1) - H2$
Runs optimization routine using MATLAB fmincon function subject to the SC routine. Centralized version uses one matrix with both elevations and velocities and decentralized version uses independent optimization routines
$u1 \leftarrow y(end - 1)$
$u2 \leftarrow y(end)$
$u1 \leftarrow max(min(u1, 0.5), 0)$ ▷ Saturates upstream gate opening
$u2 \leftarrow max(min(u2, 0.5), 0)$ ▷ Saturates middle or downstream gate opening
$v10 \leftarrow sign(dh1) * cc * u1 * sqrt(2 * g * abs(dh1))/ah1(1)$
$v1N \leftarrow sign(dh2) * cc * u2 * sqrt(2 * g * abs(dh2))/ah1(nl + 1)$
$v20 \leftarrow sign(dh2) * cc * u2 * sqrt(2 * g * abs(dh2))/ah2(1)$
$v2N \leftarrow sign(dh3) * cc * u3 * sqrt(2 * g * abs(dh3))/ah2(nl + 1)$
$v1bc = [v10\ ;\ v1\ ;\ v1N]$ ▷ Boundary velocities vector for first pool
$v1bc = [v20\ ;\ v2\ ;\ v2N]$ ▷ Boundary velocities vector for second pool
$h1p \leftarrow -(v1bc. * (A1 * h1) + fh1. * (A1 * v1bc))/dx$
$v1p \leftarrow -(g * A2 * h1 + v1. * (A2 * v1bc))/dx - g * (v1. * abs(v1). * rh1(2 : nl) - J)$
$h2p \leftarrow -(v2bc. * (A1 * h2) + fh2. * (A1 * v2bc))/dx$
$v2p \leftarrow -(g * A2 * h2 + v2. * (A2 * v2bc))/dx - g * (v2. * abs(v2). * rh2(2 : nl) - J)$
$u1p \leftarrow (-y(end - 1) + u1) * value$ ▷ Filtered input 1
$u2p \leftarrow (-y(end) + u2) * value$ ▷ Filtered input 2
$yp \leftarrow [h1p\ ;\ v1p\ ;\ h2p\ ;\ v2p\ ;\ u1p\ ;\ u2p]$ ▷ Output matrix

A.4 Countercurrent Heat Exchanger

For chapter 5, one main routine and three model routines (including constraint routine) were developed, namely:

- Permutador100 as the main routine for adaptive PWMN control for countercurrent tubular heat exchanger;
- permod100 used to define the model for centralized control on hot fluid velocity with constant cold fluid velocity;
- permod200 used to define the model for fully centralized control on hot and cold fluid velocities;
- permod300 used to define the model for non-cooperative distributed control on hot and cold fluid velocities;
- SC used to include the constraints of the optimization problem (this function is included in all model algorithms).

Algorithm 11 Permutador100 routine: Integrates different models for the heat exchanger

Begin
Defines the heat exchanger physical parameters:
L ▷ Tube length
$u \leftarrow value$ ▷ Hot fluid velocity (based upon steady-state analytical solution)
$v \leftarrow value$ ▷ Cold fluid velocity (steady-state analytical solution or constant velocity)
$a \leftarrow value$ ▷ Outer permutation coefficient
$b \leftarrow value$ ▷ Inner permutation coefficient
$W_1 \leftarrow value$ ▷ Hot fluid boundary condition
$X_0 \leftarrow value$ ▷ Cold fluid boundary condition
$nl \leftarrow value$ ▷ Number of tube sections
$dz \leftarrow L/nl$ ▷ Tube section length
$\alpha \leftarrow -L * a/v$
$\beta = -L * b/u$
$A \leftarrow (\alpha * W1 - \beta * X0 * exp(\alpha - \beta))/(\alpha - \beta * exp(\alpha - \beta))$
$B \leftarrow beta/alfa * (X0 - A)$
Calculates reference curves based on steady-state solution
$z \leftarrow 0 : dz : nl * dz$
$wref \leftarrow A + B * exp((\alpha - \beta) * z)$ ▷ Hot fluid reference
$xref \leftarrow A + \alpha/\beta * B * exp((\alpha - \beta) * z)$ ▷ Cold fluid reference
Calculates auxiliary matrices $A1$ to $B4$:
$A1 \leftarrow -eye(nl) + [zeros(nl-1,1)\ eye(nl-1,nl-1)\ ;\ zeros(1,nl)]$
$A2 \leftarrow [[zeros(1,nl-1)\ ;\ eye(nl-1,nl-1)]\ zeros(nl,1)]$
$A3 \leftarrow eye(nl) - [zeros(1,nl)\,;\, eye(nl-1,nl)]$
$A4 \leftarrow [zeros(nl-1,1)\ eye(nl-1,nl-1)\ ;\ zeros(1,nl)]$
$B1 = [1zeros(1,nl-1)]'$
$B2 \leftarrow [zeros(1,nl-1)\ 1]'$
$B3 \leftarrow B2$
$B4 \leftarrow -B1$
$I \leftarrow eye(nl)$
$x0 \leftarrow value * ones(1,nl)$ ▷ Initial cold fluid temperature
$w0 \leftarrow value * ones(1,nl)$ ▷ Initial hot fluid temperature
$V0 \leftarrow 0.5 * sum([x0 - xref(2:end)\ w0 - wref(1:end-1)] * [x0 - xref(2:end)\ w0 - wref(1:end-1)]') * dz$ ▷ Initial RCLF value based on NMPC formulation
Integrates permod100, permod200 or permod300 depending on the type of control

Algorithm 12 permod100 routine: Integrates different models for the heat exchanger

Begin
$w \leftarrow y(1 : nl)$
$x \leftarrow y(nl + 1 : 2 * nl)$
$uf \leftarrow y(2 * nl + 1)$
$n \leftarrow y(2 * nl + 2)$
$ae \leftarrow y(2 * nl + 3)$
$s \leftarrow y(2 * nl + 4)$
$be \leftarrow y(2 * nl + 5)$
$V \leftarrow 0.5 * sum([x' - xref(2 : end)\ w' - wref(1 : end - 1)] * [x' - xref(2 : end)\ w' - wref(1 : end - 1)]') * dz$
Runs optimization routine using MATLAB fmincon function subject to the constraints. Permod100 optimizes only hot fluid velocity, permod200 optimizes independently hot fluid and cold fluid velocities (distributed version) and permod300 optimizes simultaneously both velocities (fully centralized version).
$u1 \leftarrow max(min(u1, upper\_bound), lower\_bound)$ ▷ Saturates fluid velocity between lower and upper bound
$wp \leftarrow (u1/dz * A1 - I * b) * w + A2 * b * x + b * B1 * X0 + u1/dz * B2 * W1$
$xp \leftarrow (-v/dz * A3 - I * a1) * x + A4 * a1 * w + B3 * a1 * W1 - v/dz * B4 * X0$
$Vp \leftarrow sum([x' - xref(2 : end)\ w' - wref(1 : end - 1)] * [xp\ ;\ wp]) * dz$
$np \leftarrow -v/dz * (x(nl) - x(nl - 1)) + ae * (W1 - x(nl)) + value * (x(nl) - n)$
$aep \leftarrow value * (W1 - x(nl)) * (x(nl) - n)$ ▷ Value refers to the controller parameters presented in table 5.2
$sp \leftarrow u1/dz * (w(2) - w(1)) + be * (X0 - w(1)) + value * (w(1) - s)$
$bep \leftarrow value * (X0 - w(1)) * (w(1) - s)$
$ufp \leftarrow (-uf + u1)/value$ ▷ Filtered input
$yp \leftarrow [wp\ ;\ xp\ ;\ ufp\ ;\ np\ ;\ aep\ ;\ sp\ ;\ bep\ ;\ Vp]$ ▷ Output matrix