Guilherme Augusto Silva de Souza

Applicability and performance of stabilizing strategies for nonlinear model predictive controllers

São Paulo

2023

Guilherme Augusto Silva de Souza

Applicability and performance of stabilizing strategies for nonlinear model predictive controllers

Original version

Dissertation presented to the Polytechnic School of the University of São Paulo to obtain the degree of Master in Science.

Concentration area: Chemical Engineering

Advisor: Prof. Dr. Darci Odloak

São Paulo
2023

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador

São Paulo, 25 de abril de 2023

Assinatura do autor: Guilherme Augusto Silva de Souza

Assinatura do orientador

Catalogação-na-publicação

Souza, Guilherme Augusto Silva de
Applicability and performance of stabilizing strategies for nonlinear model predictive controllers / G. A. S. Souza -- versão corr. -- São Paulo, 2023.
147 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Química.

1.Controle preditivo 2.Controle de processos I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Química II.t

Universidade de São Paulo

ATA DE DEFESA

Aluno: 3137 - 10834997 - 2 / Página 1 de 1

Ata de defesa de Dissertação do(a) Senhor(a) Guilherme Augusto Silva de Souza no Programa: Engenharia Química, do(a) Escola Politécnica da Universidade de São Paulo.

Aos 28 dias do mês de março de 2023, no(a) realizou-se a Defesa da Dissertação do(a) Senhor(a) Guilherme Augusto Silva de Souza, apresentada para a obtenção do título de Mestre intitulada:

"Aplicabilidade e performance de estratégias estabilizantes para controladores preditivos baseados em modelo não-linear"

Após declarada aberta a sessão, o(a) Sr(a) Presidente passa a palavra ao candidato para exposição e a seguir aos examinadores para as devidas arguições que se desenvolvem nos termos regimentais. Em seguida, a Comissão Julgadora proclama o resultado:

Nome dos Participantes da Banca	Função	Sigla da CPG	Resultado
Darci Odloak	Presidente	EP - USP	APROVADO
Antônio Carlos Zanin	Titular	PETROBRAS - Externo	APROVADO
Márcio André Fernandes Martins	Titular	UFBA - Externo	APROVADO

Resultado Final: APROVADO

Parecer da Comissão Julgadora *

Eu, Elias Alves de Almeida ______________________________, lavrei a presente ata, que assino juntamente com os(as) Senhores(as). São Paulo, aos 28 dias do mês de março de 2023.

Antônio Carlos Zanin

Darci Odloak

Márcio André Fernandes Martins

Darci Odloak

Darci Odloak
Presidente da Comissão Julgadora

* Obs: Se o candidato for reprovado por algum dos membros, o preenchimento do parecer é obrigatório.

A defesa foi homologada pela Comissão de Pós-Graduação em ______________ e, portanto, o(a) aluno(a) ________________ jus ao título de Mestre em Ciências obtido no Programa Engenharia Química.

____
Presidente da Comissão de Pós-Graduação

Aos que se foram,
aos que vieram,
e aos que permanecem.

Acknowledgements

Agradeço ao Prof. Darci, pela liberdade e assistência dadas durante todo o mestrado.

Agradeço à minha mãe, ao meu pai, e à minha irmã, pelo apoio e paciência incondicionais.

Agradeço a todos do Bloco 21, pelas experiências compartilhadas durante cafés, almoços e hambúrgueres.

Agradeço à CAPES pelo fomento.

"A downward slope
is all I see"
Shiro Sagisu - Who Will Know

Resumo

Controle preditivo baseado em modelo tem uma extensa literatura voltada a estabilidade em malha fechada, sendo grande parte voltada a controladores baseados em modelos lineares. Faz-se uso de modelos não-lineares quando o processo químico tem não-linearidades importantes, isto é, quando a capacidade de um modelo linear em descrever a dinâmica do processo é limitada. Parte-se do mesmo princípio de adotar um horizonte infinito para que o controlador conceda estabilidade para a malha fechada. Porém, deve-se contornar a impossibilidade de computar uma trajetória infinita com um modelo não-linear. Assim, diversas estratégias foram desenvolvidas para conferir garantia de estabilidade através de provas de factibilidade recursiva e convergência da malha fechada para controladores preditivos que fazem uso de modelos não-lineares. As principais estratégias consistem em utilizar restrições terminais de igualdade, restrições terminais de desigualdade e restrições de contração. Como a aplicabilidade destes controladores não se resume à recursividade e convergência, deve-se avaliar também a capacidade do controlador em entregar ações de controle em tempo viável. Essa foi uma motivação inicial desta dissertação para a busca de estratégias estabilizantes menos restritivas que restrições de igualdade, conhecidas por demandarem consideravelmente mais esforço computacional. Assim, algumas considerações quanto à estratégias que aceleram o tempo de cômputo - estratégias de *shooting* e alternativas de integração numérica do sistema de equações diferenciais ordinárias - da solução do problema de otimização também foram feitas. Antes da implementação das formulações, um capítulo foi dedicado para melhor explorar estratégias de estabilização por retroalimentação. Essas estratégias são relevantes no escopo do trabalho por seu uso na formulação com restrição terminal de desigualdade - que depende da construção de uma região terminal invariante. Dois métodos de cálculo dessa região invariante, para sistemas em tempo discreto - foram considerados: um método baseado em um problema de desigualdades lineares matriciais (LMI, do inglês) e um problema de programação semidefinida (SDP, do inglês). Na comparação entre estes métodos, foi mostrado que o método baseado em LMI produz reguladores de ganho menor e regiões operacionais maiores, enquanto que o método baseado em SDP resulta em reguladores de ganho maior e regiões operacionais menores. Algumas formulações de controladores preditivos baseados em modelo não-linear foram implementadas, juntamente com um controlador sem elementos estabilizantes - denominado controlador sem estabilidade garantida, para comparar suas performances e requerimentos de esforço computacional. A performance das formulações foi comparada por três métricas: distância da zona de controle, esforço de controle e distância de alvo econômico, considerando um sistema não-linear que representa uma associação de quatro tanques. No exemplo do tanque quádruplo em malha fechada com as formulações mostradas, no quesito de distância da zona de controle a restrição terminal de desigualdade demonstrou a melhor performance, enquanto que a formulação sem estabilidade garantida mostrou a pior performance. O menor esforço de controle foi observado pela formulação sem estabilidade garantida, enquanto que o maior esforço de controle foi exercido pela formulação com restrição terminal de contração.

Quanto a distância do alvo econômico, ambas as formulações sem estabilidade garantida e com restrição terminal de igualdade obtiveram a pior performance, sendo a melhor performance econômica da formulação com restrição terminal de desigualdade. Como a formulação com restrição terminal de igualdade demandou maior esforço computacional no exemplo mencionado, essa formulação foi então adaptada para acelerar a velocidade de convergência da otimização por meio de alteração de seus métodos de *shooting* e de integração numérica. Com essas alterações, a demanda de esforço computacional dessa formulação foi reduzida a ponto de ter demandar esforço computacional comparável a todas as outras formulações. Fez-se também uma análise de sensibilidade dos parâmetros de sintonia para todas as formulações, e suas performances em malha fechada considerando estes índices de performance citados foi avaliada mais uma vez.

Palavras-chave: Controle preditivo, controle de processos, controle não-linear, estabilidade de controle

Abstract

Model predictive control has extensive literature on closed-loop stability, with most of it dealing with predictive controllers based on linear models. Nonlinear models are used when the chemical process has considerable nonlinearities which cannot be properly described by linear models. Infinite horizon controller is the starting point for both linear and nonlinear model predictive controllers, with the latter being impossible to compute. Multiple strategies were developed in order to grant closed-loop stability via recursive feasibility and closed-loop convergence. Main stabilizing strategies consist of utilizing terminal equality constraints, terminal inequality constraints and terminal contracting constraints. However, the applicability of such controllers does not only rely on feasibility and convergence but these formulations must be able to deliver, in proper time, the next control actions. This was an initial motivation of this dissertation. Stabilizing strategies less restrictive than equality constrained formulations were researched. Before implementing NMPC formulations, feedback stabilization strategies are explored in a separate chapter - since these are relevant in the construction of an invariant region for a controller formulation dependant on terminal regions. Two discrete-time strategies of computing said invariant regions are compared in a closed-loop simulation, LMI-based and SDP-based operating regions. The trade-off between closed-loop performance and operating region size is shown, with LMI-based methods favoring operating region size and SDP-based methods producing regulators with larger gain matrices. Some stabilizing strategies were implemented, along with a controller without stabilizing elements - named no guaranteed stability controller - in order to compare their performance and computational effort demands. Their performances were compared in three aspects: distance-to-zone, control effort and distance to economic target, considering these formulations were in closed-loop with a nonlinear system representing a quadruple-tank system. In the quadruple-tank example and regarding distance-to-zone, best and worst formulations were respectively terminal inequality constrained formulation and formulation without guaranteed stability. Terminal contracting constraint formulation exerted the most control effort while the least control effort was performed by the formulation without guaranteed stability. Economic performance of the terminal inequality constraint formulation was the best, while the worst performance was close between terminal equality constraint formulation and formulation without guaranteed stability. On computational effort, as expected the terminal equality constrained showed the highest average computer time. This formulation was modified in order to improve its convergence speed, with changes to its shooting method and numerical integration. With these changes, computational effort was sufficiently reduced such that its convergence speed became comparable with all other formulations deployed. Sensitivity analysis of the tuning parameters of each formulation was executed as well for all of the deployed formulations, and the effect of changes in tuning parameters on closed-loop performance indexes proposed was evaluated once again.

Keywords: Predictive control, process control, nonlinear control, control stability.

List of Figures

List of Tables

List of abbreviations and acronyms

CSTR	Continuously-stirred tank reactor
DARE	Discrete-time algebraic Riccati equation
FHOCP	Finite horizon optimal control problem
IHOCP	Infinite horizon optimal control problem
LMI	Linear matrix inequality
LQR	Linear quadratic regulator
MPC	Model predictive control
NC	No constraint
NLP	Nonlinear programming
NMPC	Nonlinear model predictive control
QIH	Quasi-infinite horizon
QP	Quadratic programming
SDP	Semi-definite programming
TCC	Terminal contracting constraint
TEC	Terminal equality constraint
TIC	Terminal inequality constraint

List of symbols

Symbol	Description (first occurrence)
$\boldsymbol{x}$	state (2.1)
t	time (2.1)
$\dot{\boldsymbol{x}}$	state differential w.r.t. time (2.1)
$\boldsymbol{f}(\cdot)$	function (2.1)
$\boldsymbol{u}$	parameter or input (2.1)
$\mathscr{X}$	set of admissible states (2.1)
$\mathbb{R}^{n_x}$	n_x-dimensional Euclidean space (2.1)
$\mathscr{U}$	set of admissible inputs (2.1)
$\boldsymbol{A}$	state matrix (2.4)
$\boldsymbol{B}$	input matrix (2.5)
$\mathscr{X}_\omega$	limit set (Definition 2)
∞	infinity (2.7)
M	smooth manifold (Subsection A.2)
C	curve defined in a smooth manifold M (Subsection A.2)
n	sequence counter (2.7)
$\lVert\cdot\rVert$	norm (2.11)
C_L	Lipschitz constant (2.11)
r	radius of a ball B_r around a point (2.12)
B_r	ball of radius r around a point (2.12)
c	base constant (2.13)
$\mathbb{R}$	set of real numbers (Subsection A.2)
I	subinterval of $\mathbb{R}$ (Subsection A.2)

$\boldsymbol{v}$	tangent vector to a curve C (Subsection A.2)
$\|_{\boldsymbol{x}}$	defined at a point $\boldsymbol{x}$ (Subsection A.2)
$TM\|_{\boldsymbol{x}}$	tangent space to a point $\boldsymbol{x} \in M$ (Subsection A.2)
G	group (Subsection A.4)
$m(\cdot,\cdot)$	group multiplication (Subsection A.4)
n, o, p	elements of a group (Subsection A.4)
Id	identity element of a group (Subsection A.4)
n^{-1}	inverse element of a group (Subsection A.4)
$[\cdot,\cdot]$	Lie bracket, group multiplication of a Lie group (Subsection A.5)
h_{ij}^{k}	smooth real function (Subsection A.6)
$\mathfrak{g}$	Lie algebra of a Lie group G (Subsection A.6)
N	submanifold (Subsection A.6)
H	local Lie group (Subsection A.6)
$\mathfrak{h}$	local Lie algebra (Subsection A.6)
$J(\boldsymbol{x},\boldsymbol{u})$	objective functional (4.1a)
$l(\boldsymbol{x}(t),\boldsymbol{u}(t))$	stage functional (4.1a)
$(k+j\|k)$	value at j sampling intervals forward (discrete time) (4.2d)
$\boldsymbol{Q}_{\boldsymbol{x}}$	error penalty matrix (4.2)
$\boldsymbol{R}$	control move penalty matrix (4.2)
N_p	prediction horizon (4.2)
N_c	control horizon (4.2)
$\boldsymbol{u}_k$	input trajectory at time k (4.2)
$\boldsymbol{Q}_u$	economic penalty matrix (4.5)
$\boldsymbol{S}$	terminal slack penalty matrix (4.16)
$\boldsymbol{W}$	contracting factor penalty matrix (4.16)

Quantifiers and binary operators

$\forall$	for all (2.1)
$\geq$	greater than or equal (2.1)
$:=$	by definition (2.4)
$\leq$	less than or equal (2.11)
$\in$	is an element (Section 2)

Subscripts and superscripts

eq	equilibrium related (2.3)
ω	limit (Definition 2)
0	at time $t = 0$ (4.1e)
t_1	at time $t = t_1$ (4.1f)
$*$	optimal (4.1a)
$terminal$	terminal (4.1a)
sp	set point (4.2)
z	control zone (4.5)
des	desired input value (4.5)
tar	target input value (4.15)
min	minimum value (4.16)
$\bar{\cdot}$	mean value (Table 5)

Greek letters

λ	eigenvalue (2.6)
ω	omega (see Subscripts and superscripts)
σ	exponential constant (2.13)
ψ	flow of a dynamic system (Subsection A.3)

ϕ	parametrization of a curve C in a manifold M (Subsection A.2)
γ_i	i-th parameter of CSTR model (Subsection 2.2.1)
$\sum$	summation (4.2)
$\boldsymbol{\Delta u}$	control move (4.2e)
$_{\Delta}$	related to control moves (4.2h)
ε	steady state tolerance (Subsection 4.2.1)
δ	terminal slack value (4.16)
α	contraction factor (4.16)
ς	standard deviation (Table 5)

Contents

1 Introduction

In process systems, a process that operates autonomously and delivers product of consistent quality is most desirable. One would develop a process that allows products of different qualities through manipulation of certain process parameters, with said manipulation being handled by a controller. The controller manipulates these process parameters, named inputs, in order to bring product quality, determined by outputs, to a desired value, named set point.

Through successive solutions of online optimal control problems, model predictive control (MPC) computes the control actions to be taken in order to reach its specified objective. A common objective given to an MPC is set point tracking, sometimes control move smoothing (for actuator preservation) and some economy-related such as product maximization or minimal expenditure of resources. The controller uses a given model in order to predict the future values of outputs and compute control actions that satisfy its given objective.

MPC is classified by the mathematical model it uses for output or state prediction: linear or nonlinear (NMPC). Earlier applications of MPC used linear models with some sort of circumvention of constraints - formulations such as LDMC (MORSHEDI; CUTLER; SKROVANEK, 1985) and QDMC (GARCIA; MORSHEDI, 1986), as solving nonlinear programming (NLP) problems - result of constrained MPC schemes - lead to prohibitive computational times to be used online. Recently there were advances to solve these types of problems, along with a desire for better product quality, operational safety, and economic gains, so nonlinear models have received more attention industry-wise, although there were NMPC formulations (ECONOMOU; MORARI; PALSSON, 1986) studied along with linear MPC.

When in closed-loop, stability leads to product quality of little to no qualitative or quantitative variation, as well as theoretically rendering process runoffs impossible. With this argument, stability should be an important characteristic of any deployed formulation.

However, product quality cannot be achieved by use of a controller with a rigorous nonlinear model alone, as the controller may not deliver control actions in proper time. Recursive feasibility and convergence are desirable theoretical properties but in practice, computational effort demanded by the formulation must be compatible with process sampling time and available computational power for safe closed-loop operation.

1.1 Motivation

As the algorithms for solving nonlinear programming problems improve, model predictive control can use nonlinear models in its predictions. These models may better represent process systems in chemical engineering, as first-principle models are often used in process synthesis. Prior to implementing stabilizing strategies, an NMPC without any stabilizing element was implemented. In its simulation results, it has been observed that nonlinear model predictive controllers demonstrate behavior of guaranteed stable controllers - namely nonincreasing cost function - without any stabilizing elements in its formulation. Understanding the conditions that grant stability in this case was one driving force in the production of this work.

Most of the stabilizing strategies in the literature were developed when online solving of nonlinear programming problems (NLPs) was nearly impossible. Due to minute computational power, solution times incompatible with sampling times of process systems. The industry has not yet reached abundance in computational power, which calls for cautious evaluation whether a formulation could be implemented online. Regardless of theoretical results, this work was concerned with theoretically sound formulations while attempting to lower computational effort.

Finally, this work was designed as an effort to expand comparisons between stabilizing formulations. Unfortunately, literature in stabilizing formulations is often concerned with a novel stabilizing strategy but rarely bringing a comparison with other stabilizing strategies. In addition to that, since theoretically sound formulations will operate with any tuning parameters (error penalties, horizon lengths), controller tuning - performance-wise a very important topic - is rarely discussed.

1.2 Objectives

This work seeks to demonstrate some of the stabilizing strategies found in NMPC literature, adapt said formulations to operate with control zones and optimizing targets, and develop a method that compares closed-loop performance of said controllers.

1.2.1 Specific objectives

- Expose different concepts of equilibrium from nonlinear system theory as well as dynamic system theory and show the evolution from optimal control problems to finite horizon optimal control problems;
- Expose, implement multiple stabilizing NMPC formulations from the literature, and compare closed-loop performance with different performance metrics;
- Evaluate controller tuning effects on closed-loop performance of different stabilizing strategies.

1.3 Methodology

This work consists of *in silico* simulations of mathematical models represented by ordinary differential equations, operating in closed-loop with optimization problems that represent nonlinear model predictive controllers. The optimization problems were programmed with CasADi (ANDERSSON et al., 2019) in MATLAB (THE MATHWORKS, INC., 2017). Numerical integrations of said differential equations were performed with CVODES solver from the SUNDIALS suite (HINDMARSH et al., 2005). Quadrature procedures for the differential equations were performed with CasADi functions as well, with Gauss-Radau polynomial with three appending points used. The optimization algorithm used to solve the nonlinear programming problems was IPOpt, an interior-point filter line-search algorithm (WÄCHTER; BIEGLER, 2006). The linear matrix inequality (LMI) problems were programmed with YALMIP (LÖFBERG, 2004) in MATLAB, with an additional algorithm for semi-definite programming from SDPT3 (TÜTÜNCÜ; TOH; TODD, 2003) due to its capability of handling logarithms of determinants. Semidefinite programming problems (SDP) were also programmed with YALMIP, and solved via SeDuMi-1.3 (STURM, 1999). The computer used in the *in silico* simulations has an AMD Ryzen 7 5700X @ 3.4GHz, with its Matlab built-in benchmark times of $\begin{bmatrix} 0.1548 & 0.0439 & 0.0294 & 0.1246 & 0.1733 & 0.2584 \end{bmatrix}$ computed via `benchmark` function.

1.4 Organization of the dissertation

This dissertation consists of four parts. Part II approaches nonlinear systems theory as well as concepts utilized in the equilibrium point studies performed in this area. Part III approaches feedback stabilization techniques of nonlinear systems. Part IV exposes NMPC formulations and their closed-loop convergence capabilities, as well as some discussion regarding computational effort of each formulation. Part V contains most of the closed-loop results of some of the NMPC formulations presented in the previous part, as well as closed-loop performance and computational effort comparisons between deployed formulations. Then, conclusions obtained from each part are recalled, as well as lines of work that can originate from this work are discussed.

Part I

Nonlinear systems

2 Nonlinear systems

A mathematical model composed of equations that contain nonlinearities (relationships between variables that cannot be globally approximated by linear combinations of these variables) is named a nonlinear system. Its most generic form is

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \forall\, t \geq 0, \tag{2.1}$$

where the relations between time t, the states $\boldsymbol{x} \in \mathscr{X} \subset \mathbb{R}^{n_x}$ and other parameters $\boldsymbol{u} \in \mathscr{U} \subset \mathbb{R}^{n_u}$, represented by the mapping $\boldsymbol{f} : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, may or may not be necessarily nonlinear. Time-invariant systems have no parameters that are time dependent, whereas time-variant systems do have such time dependencies. This work will focus on time-invariant systems, such as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \forall\, t \geq 0. \tag{2.2}$$

Notice how time is no longer an argument of the mapping $\boldsymbol{f}$.

A central definition of nonlinear systems analysis is the concept of *equilibrium*.

Definition 1 *A point $(\boldsymbol{x}_{eq}, \boldsymbol{u}_{eq})$, $\boldsymbol{x}_{eq} \in \mathscr{X} \subset \mathbb{R}^{n_x}$, $\boldsymbol{u}_{eq} \in \mathscr{U} \subset \mathbb{R}^{n_u}$, is said to be an equilibrium point of a system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ if:*

$$\boldsymbol{f}(\boldsymbol{x}_{eq}, \boldsymbol{u}_{eq}) = \boldsymbol{0} \tag{2.3}$$

The equilibrium points can be found by open-loop simulations of the system, via root finding algorithms or by parametric continuation methods (DHOOGE et al., 2008). They can be classified by their stability by evaluating the system's eigenvalues at that specific point. The eigenvalues λ are computed from the Jacobian linearization of the nonlinear system at said point.

$$\boldsymbol{A} := \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}_{eq}, \boldsymbol{u}_{eq})} \tag{2.4}$$

$$\boldsymbol{B} := \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{x}_{eq}, \boldsymbol{u}_{eq})} \tag{2.5}$$

$$\lambda = \operatorname{eig}(\boldsymbol{A}) \tag{2.6}$$

Regarding models defined in continuous-time, eigenvalues with negative real parts mean the equilibrium point is stable. Eigenvalues with non-negative real parts mean that the equilibrium point is unstable. Eigenvalues with imaginary parts indicate oscillatory behavior.

As for models in discrete-time, eigenvalues contained in the unit circle are stable, otherwise the model is unstable.

We follow the definition of an equilibrium point to the definition of the set of equilibrium points.

Definition 2 *A ω-limit set, denoted by $\mathscr{X}_\omega$, is the set of all equilibrium points for positive infinite time. For a nonlinear system* (2.1) *we have*

$$\lim_{n\to\infty} t_n = +\infty \tag{2.7}$$

$$\lim_{n\to\infty} \boldsymbol{f}(t_n, \boldsymbol{x}(t), \boldsymbol{u}_{eq}) = \boldsymbol{0} \tag{2.8}$$

$$\lim_{n\to\infty} \boldsymbol{x}(t_n) = \boldsymbol{x}_{eq} \tag{2.9}$$

Regarding stability of trajectories, evaluating eigenvalues does not contribute to stability analysis since it is valid for one point only instead of the whole trajectory. Lyapunov stability theory, to be exposed in Section 2.1, offers strong tools that enable determining stability of a system, be it a closed-loop or open-loop system.

2.1 Stability in the sense of Lyapunov

Stability evaluation of nonlinear systems consist of evaluating its functions instead of point-wise eigenvalue. Consider the following system of differential equations that describe the evolution of the states $\boldsymbol{x}$ with $\boldsymbol{u}$ as parameters:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.10}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ and $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$, $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is continuous. We also assume that $\boldsymbol{f}$ is globally continuous in Lipschitz, which means it satisfies a Lipschitz condition such as:

$$\|\boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{u}) - \boldsymbol{f}(\boldsymbol{x}_2, \boldsymbol{u})\| \leq C_L \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|, \quad C_L > 0. \tag{2.11}$$

where $\|\cdot\|$ stands for a norm. Some authors refer to the smallest value of C_L that attends to condition (2.11) as the Lipschitz constant.

Lyapunov's direct method revolves around the time derivative of $\boldsymbol{f}$. This method has received this denomination because it handles the differential equation directly, instead of working with its solutions. It is also referred to as the second method in the literature. Lyapunov methods that evaluate solutions of differential equations have received the denomination of (Lyapunov) indirect methods.

Theorem 1 *(VIDYASAGAR, 1993) An equilibrium point* $\mathbf{0}$ *is said to be stable in the sense of Lyapunov if there is a continously differentiable, locally positive definite function* $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ *and a constant* $r > 0$ *such that:*

$$\frac{\partial \boldsymbol{f}}{\partial t} \leq 0,\ \forall t \geq t_0,\ \forall \boldsymbol{x} \in B_r \tag{2.12}$$

Theorem 1 says that a candidate Lyapunov function $\boldsymbol{f}$ is stable in an open ball B_r around the equilibrium point $\mathbf{0}$ if its time derivative shows nonpositive behavior. When condition 2.12 is satisfied, then the candidate Lyapunov function is named Lyapunov function.

Another type of stability is exponential.

Definition 3 *A system is said to be exponentially stable if there are two positive constants* c, σ *where the following condition holds:*

$$||\boldsymbol{x}|| \leq ce^{-\sigma t} \tag{2.13}$$

Meaning that if one can show that the evolution of the norm $||\boldsymbol{x}||$ over time is bounded from above by an exponential decay function with parameters c and σ, the system is exponentially stable.

For model predictive control, one usually evaluates the objective function in order to identify nonincreasing behavior observed by Lyapunov functions. Such behavior indicates closed-loop stability. Stability classification then depends on the characteristics of the bounding functions used to show stability: asymptotic bounding functions yield asymptotic stability and exponentially decaying bounding functions yield exponential stability.

2.2 Dynamic systems

In this section a different approach to dynamic system analysis is given. The intent of this approach is to enable studies on stability and controllability of dynamic systems in the same way as it has been done throrougly in linear systems. The first application of differential geometry was done by Sussmann & Jurdjevic (1972), where the discussion of controllability for nonlinear systems was initiated.

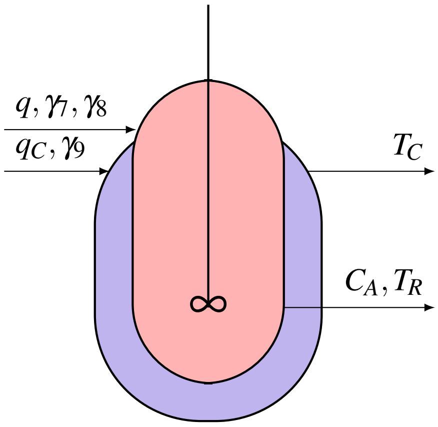
Some concepts of differential geometry are exposed in Appendix A, regarding manifolds, (geometric) flows and the tools necessary to establish equilibrium points from another perspective, such as Lie brackets and Lie algebras. The main result in this work regarding differential geometry is enabling the computation of equilibrium points via commutativity of vector fields that define a dynamic system, meaning the endpoints of a flow are equal - the dynamic system does not change state.

Flow endpoints meeting can be evaluated point-wise by checking for linear independence of vector fields at that point. When they are linearly dependent, they are incapable of generating non-zero flow. In dynamic systems, one usually wishes to determine either a vector field which enables access to a part of the state space (for example, a vector field which would enable one vehicle to access a specific direction) or which vector fields one should add to completely eliminate any kind of dynamic (namely, an annihilator).

2.2.1 Case study: jacketed reactor

Application of meeting endpoints was done in Souza et al. (2021), where we compute the static operability region of a plant via Lie algebra, instead of computing the steady state directly as a solution of a system of ordinary differential equations which represents the plant. In this work, the following dynamic system - which represents a jacketed continuously-stirred tank reactor (CSTR) - was studied. The dynamic system was approached in the work of Russo & Bequette (1995). The system in question can be visualized in Figure 1.

Figure 1 – Diagram of a jacketed CSTR



Source: Own author.

where the reactor and the jacket have their own inlets and outlets. The dynamic system is as follows:

$$\begin{bmatrix} \dot{C}_A \\ \dot{T}_R \\ \dot{T}_C \end{bmatrix} = \underbrace{\begin{bmatrix} -\gamma_6 C_A \exp\left(\frac{T_R}{1+\frac{T_R}{\gamma_2}}\right) \\ -\gamma_3(T_R - T_C) + \gamma_1\gamma_6 C_A \exp\left(\frac{T_R}{1+\frac{T_R}{\gamma_2}}\right) \\ \gamma_4\gamma_3\gamma_5(T_R - T_C) \end{bmatrix}}_{f} + \underbrace{\begin{bmatrix} \gamma_7 - C_A \\ \gamma_8 - T_R \\ 0 \end{bmatrix}}_{g_1} q + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \gamma_4(\gamma_9 - T_C) \end{bmatrix}}_{g_2} q_C, \quad (2.14)$$

where there are three ordinary differential equations: dimensionless mass and energy balances for the reactor and a dimensionless energy balance for its jacket. The states C_A, T_R and T_C represent respectively the reactor output reactant concentration ratio, reactor output temperature ratio and jacket output temperature ratio. Inputs q, q_C represent respectively the reactor feed flow ratio and the jacket feed flow ratio. The parameters γ_i, $i = 1, \ldots, 9$, correspond to case 2 (RUSSO; BEQUETTE, 1995):

$$\gamma = \begin{bmatrix} 8.0 & 20.0 & 0.3 & 10 & 1 & 0.072 & 1 & 0 & -1 \end{bmatrix}.$$

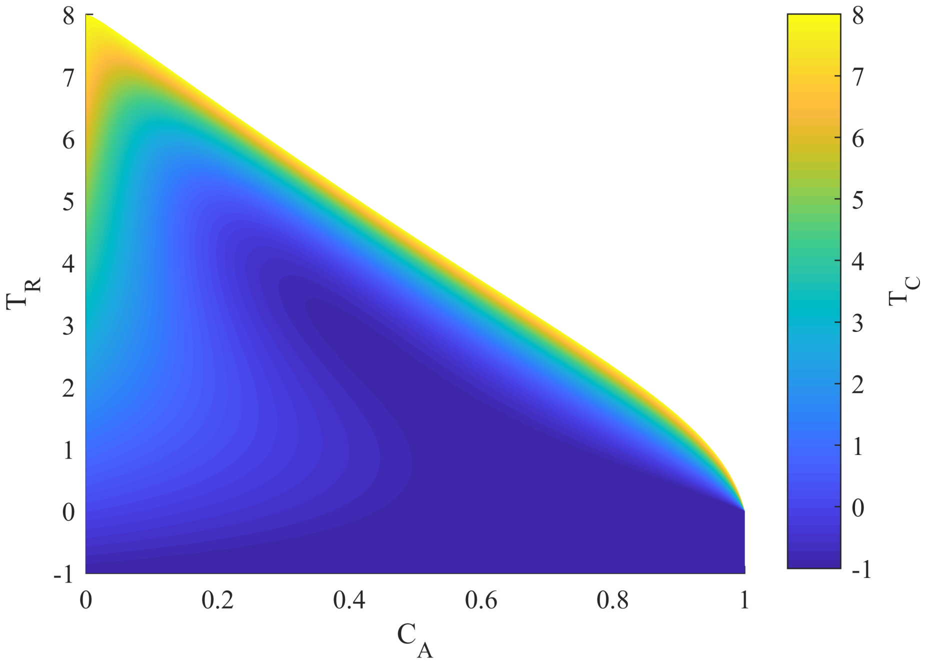
With these parameters, the system shows an interesting behavior, given by an ignition state and an extinction state. Due to the presence of these modes, inbetween there lies an unstable mode.

The static operability region of the reactor is computed. An alternative definition of this region is a subset of the ω-limit set of the dynamic system. This subset is determined by equation (2.15):

$$\det \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) & \boldsymbol{g}_1(\boldsymbol{x}) & \boldsymbol{g}_2(\boldsymbol{x}) \end{bmatrix} = 0, \tag{2.15}$$

with the state $\boldsymbol{x}$ as the unknown. The determinant equation (2.15) is used to determine states $\boldsymbol{x}$ where the vector fields $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{g}_1(\boldsymbol{x})$, and $\boldsymbol{g}_2(\boldsymbol{x})$ commute, that is, when they have meeting endpoints or are linearly dependent, as described earlier. States that solve such equation can be seen in Figure 2.

Figure 2 – Unstable CSTR static operability surface. Colormap on the right represents jacket temperature ratio.



Source: Own author.

This surface can be obtained by writing the determinant equation (2.15) with the Symbolic Math Toolbox in MATLAB (THE MATHWORKS, INC., 2017), then using the plotting command *surf*.

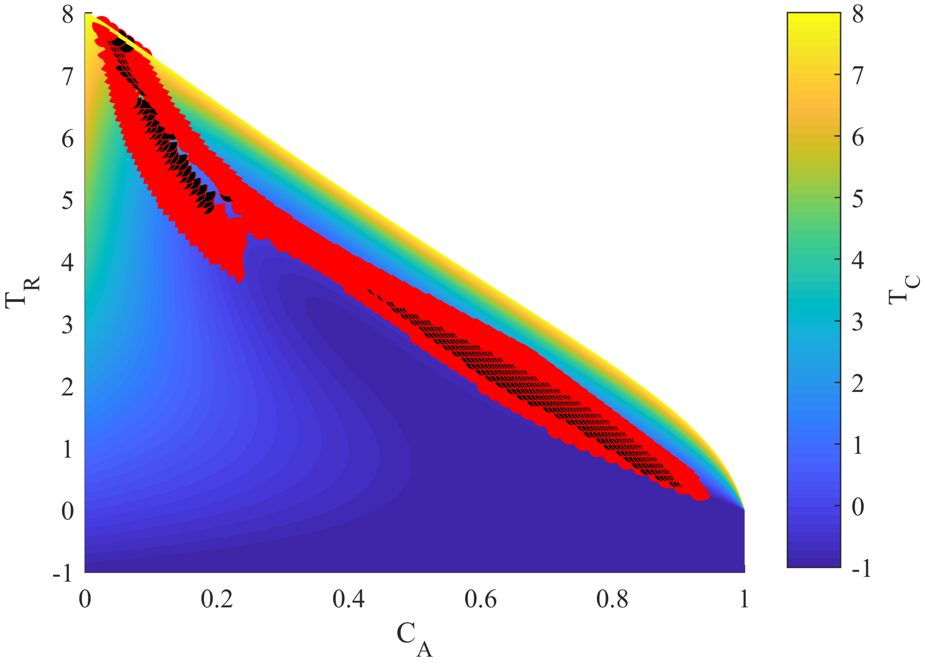
From Figure 2, one could affirm that this reactor can be maintained at every reactant conversion. However, this cannot be concluded without further information about inputs required to attain such equilibria.

The next step of the method is to compute the corresponding inputs to the operating regions one would choose in Figure 2. These inputs are computed as roots for the dynamic system itself, after substitution of the points that are elements to ω-limit set. Assuming the following admissible and desired input sets:

$$\mathcal{U}_{adm} = \left\{ \boldsymbol{u} \in \mathbb{R}^{n_u} \middle| \begin{bmatrix} 0.5 & 0 \end{bmatrix}^T \le \boldsymbol{u} \le \begin{bmatrix} 1.5 & 2 \end{bmatrix}^T \right\},$$
$$\mathcal{U}_{des} = \left\{ \boldsymbol{u} \in \mathbb{R}^{n_u} \middle| \begin{bmatrix} 0.8 & 0 \end{bmatrix}^T \le \boldsymbol{u} \le \begin{bmatrix} 1.2 & 1 \end{bmatrix}^T \right\}.$$

Figure 3 marks equilibria that are attained with admissible and desired inputs chosen.

Figure 3 – CSTR static operability surface with admissible (●) and desired inputs (●). Colormap on the right represents jacket temperature ratio.

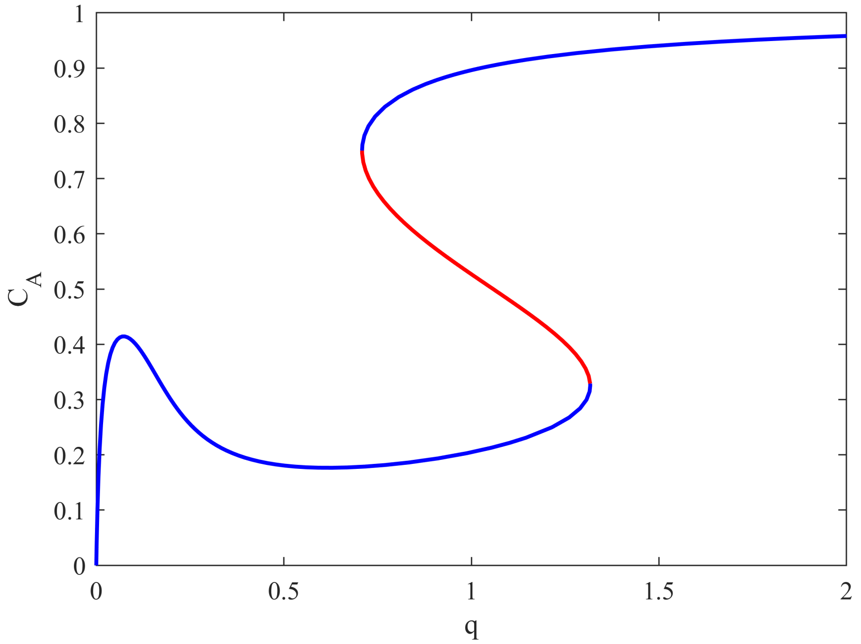


Source: Own author.

The admissible input set and desired input set realize limit points for high and low conversion operating regions. This information serves as a guide in choosing operating zones for zone control, as well as indicating in process synthesis step whether operability regions are feasible or not. As an example, one can conclude from Figure 3 that conversion of 40%, $C_A = 0.6$, with reactor temperature ratio $T_R = 1$ is a possible static operating point, but unrelated to admissible inputs.

Further understanding of this system is done via parametric continuation of equilibrium points, performed via MatCont (DHOOGE et al., 2008) in MATLAB (THE MATHWORKS, INC., 2017). Parametric continuation requires a degree of freedom in order to compute close equilibria "forward" or "backward". Figure 4 illustrates an example of MatCont's output.

Figure 4 – Output reactant ratio C_A by feed flow ratio q for nominal jacket feed flow ratio $q_c = 1$, with stable (—) and unstable equilibria (—)



Source: Own author.

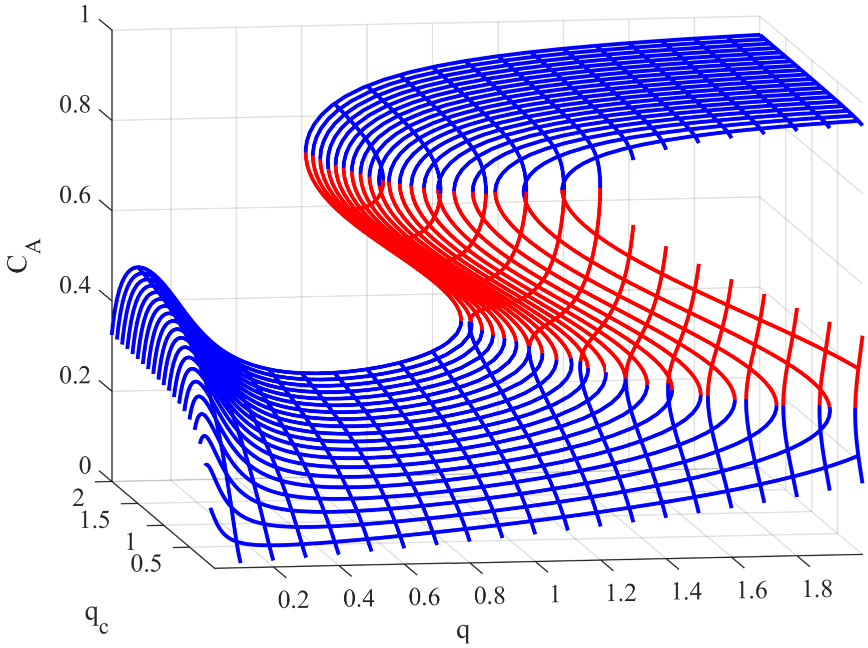
MatCont not only computes equilibria by varying the chosen degree of freedom, q in Figure 4, but is capable of classifying equilibria via eigenvalues. Determining unstable equilibria via open-loop simulations is nearly impossible since the initial guess has to be exact - otherwise the simulation evolves towards a stable equilibrium. Parametric continuation in the context of process systems engineering is useful for determining equilibria with interesting product qualities that are difficult to determine via open-loop simulations.

For this system, if one were interested in finding a static operating point of intermediary conversion with nominal reactor feed flow ratio and nominal jacket feed flow ratio, Figure 4 shows that there exists such point, whereas open-loop simulations would yield either equilibria in the ignition zone of high conversion or extinction zone of low conversion.

Figure 4 also shows output multiplicity as well as input multiplicity. In chemical engineering, it is known that reactive systems such as reactors (RUSSO; BEQUETTE, 1995) and reactive distillation columns (KUMAR; KAISTHA, 2008) may show output and input multiplicity. For this system, if one were to draw a vertical line, one can find up to three different outputs for the same input - output multiplicity, whereas horizontal line drawn would show three different inputs for the same output - input multiplicity.

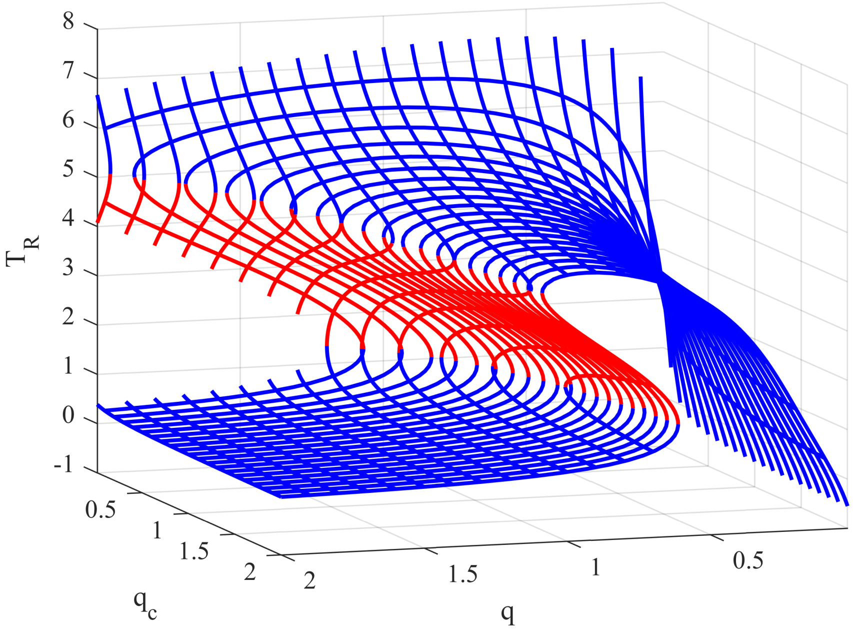
If one were to compute these curves for multiple fixed values for jacket feed flow ratio with free reactor feed flow ratio, as well as multiple fixed values for reactor feed flow ratio with free jacket feed flow ratio, these curves show a surface equilibria, now relating an output by both inputs. Figure 5 shows the output reactant ratio by both reactor and jacket feed flow ratio. Figure 6 the output temperature ratio T_R equilibrium surface is shown.

Figure 5 – Output reactant ratio C_A by feed flow ratio q and jacket feed flow ratio q_c, with stable (——) and unstable equilibria (——)



Source: Own author.

Figure 6 – Output temperature ratio T_R by feed flow ratio q and jacket feed flow ratio q_c, with stable (——) and unstable equilibria (——)

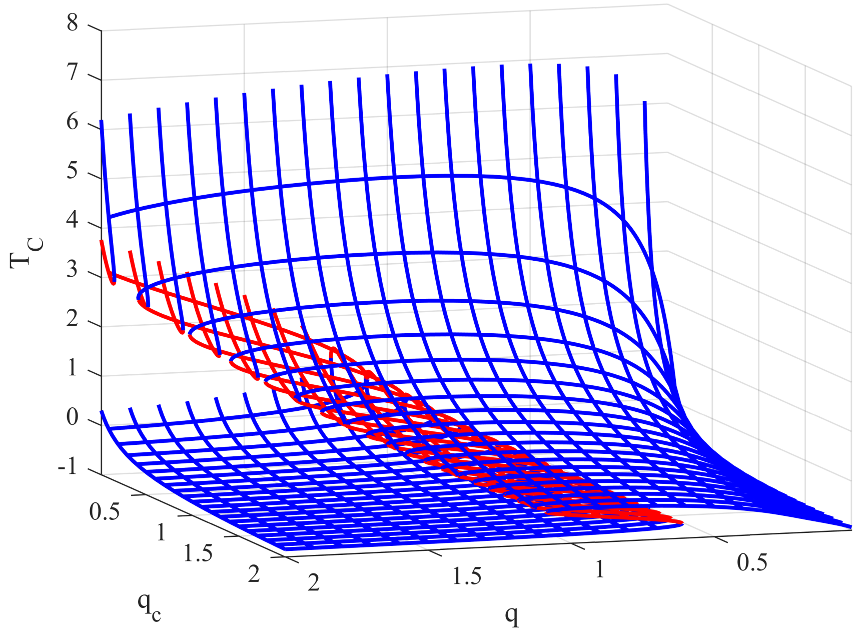


Source: Own author.

Figures 5 and 6 show that the reactor presents an entire unstable operating region of intermediary conversion (approximately $C_A = [0.4,\ 0.7]$). It is seen in detail as well the extinction and ignition modes of the reactor. The high temperature ratio surface in Figure 6 is connected to the low concentration ratio surface of Figure 5, since the ongoing reaction is exothermic. The correspondence is evident from the input values as well.

For completion's sake, Figure 7 contains the equilibrium surface for the jacket temperature ratio.

Figure 7 – Output jacket temperature ratio T_C by feed flow ratio q and jacket feed flow ratio q_c, with stable (—) and unstable equilibria (—)

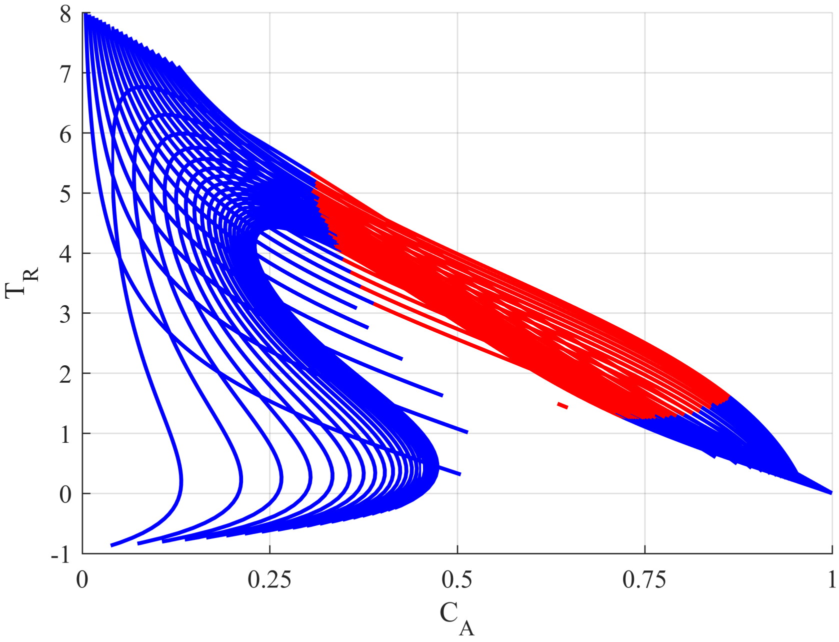


Source: Own author.

This surface is similar to the reactor temperature ratio because of the thermodynamic equilibrium between the reactor and the jacket. Notice as well that as the jacket feed flow ratio q_c increases, the temperature difference between the reactor and the jacket increases. With excess coolant used, the jacket temperature tends to not change as the coolant removes heat from the reactor. This is the main difference between the two temperature surfaces, the jacket temperature ratio surface becomes more "flattened" as the jacket feed flow ratio increases.

Finally, the static operability region in the state space can be seen in Figure 8.

Figure 8 – Output reactor temperature ratio T_R by output reactant ratio C_A, with stable (—) and unstable equilibria (—)

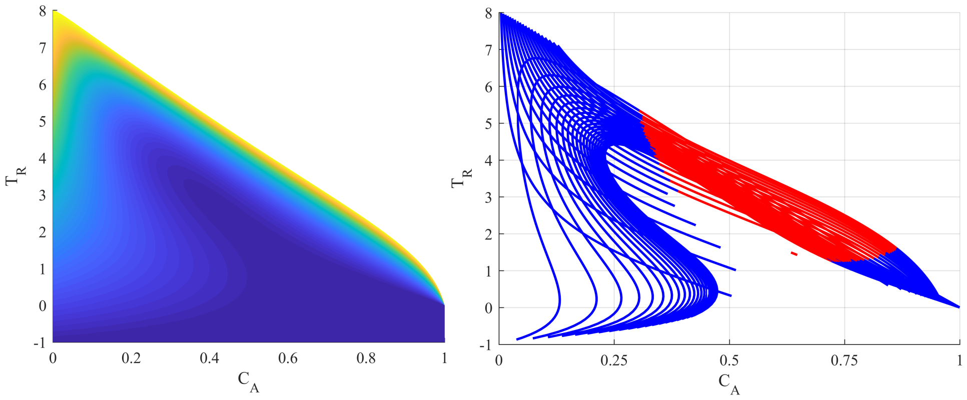


Source: Own author.

Figure 8 shows an interesting detail that was not present in Figure 2, the location of the unstable equilibrium region. A limitation of the parametric continuation is also seen: the surface is not complete due to the chosen cut-off interval of input variation, that was $[0, 2.0]$ for both inputs. Most of the missing surface at the bottom of the figure correspond to high values of inputs - not contemplated by the interval chosen.

For comparison, the static operability regions determined by both sets are compared.

Figure 9 – Static operability regions determined by determinant equation (2.15) (left) and parametric continuation (right)



Source: Own author.

The input cut-off for the parametric continuation method limits the method's applicability over the entire state space, but most of the state space information is useless since very high values of inputs in this case are impractical: the reactor or jacket would not be designed to support over two times its nominal feed flow.

These methods were useful in gathering equilibria information of the system in question. Stabilizing MPC strategies often revolve around an equilibrium point, so strategies of gathering such points are useful, especially when these equilibria are not accessible by open-loop simulations.

Part II

Feedback stabilization

Feedback stabilization is a commonly studied control problem in the literature. In this work, the linear quadratic regulator (LQR) will be explored in discrete-time applications. Some formulations require the development of LQR in continuous-time, such as the work of (CHEN; ALLGÖWER, 1998). However, since continuous-time formulations are of little interest for industrial applications, said formulations will not be shown in this work. The discrete-time LQR is used as a virtual controller, enabling boundedness of the "tail" of the cost function of an NMPC formulation. Before it is deployed along with an NMPC formulation in a latter part of this work, the discrete-time regulator is better explored on its own.

The methods of obtaining an LQR in this work consist of a traditional and a more recent approach found in the literature. These methods were extended to account for bounded input moves. These modifications enhance applicability of the methods found in the literature, as controllers synthesized with input move bounds have more general applications in the industry.

3 Discrete-time feedback stabilization

Feedback stabilization of a discrete-time nonlinear system will be discussed. We begin with a nonlinear system such as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{3.1}$$

and the corresponding Jacobian linearization, $(\boldsymbol{A}_{continuous}, \boldsymbol{B}_{continuous})$, obtained on an equilibrium point, $(\boldsymbol{x}_{sp}, \boldsymbol{u}_{sp})$, that can be determined by discretizing a continuous-time Jacobian linearization such as

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}_{continuous}(\boldsymbol{x}(t) - \boldsymbol{x}_{sp}) + \boldsymbol{B}_{continuous}(\boldsymbol{u}(t) - \boldsymbol{u}_{sp}) \\ &= \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}_{sp}, \boldsymbol{u}_{sp})}(\boldsymbol{x}(t) - \boldsymbol{x}_{sp}) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{x}_{sp}, \boldsymbol{u}_{sp})}(\boldsymbol{u}(t) - \boldsymbol{u}_{sp}),\end{aligned}$$

with a command such as MATLAB's *c2d*. The linearized model after discretization would be of the form

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(\boldsymbol{x}(k) - \boldsymbol{x}_{sp}) + \boldsymbol{B}(\boldsymbol{u}(k) - \boldsymbol{u}_{sp}), \tag{3.2}$$

which is a linear difference equation. Notice that the models (3.1) and (3.2) are not the same, since the former can be simply a nonlinear system, whereas the latter is a linear model with state gain and input gain matrices $(\boldsymbol{A}, \boldsymbol{B})$. One can determine a gain matrix $\boldsymbol{K}$ such that a control law

$$\boldsymbol{u}(k) = \boldsymbol{u}_{sp} - \boldsymbol{K}(\boldsymbol{x}(k) - \boldsymbol{x}_{sp}) \tag{3.3}$$

is stabilizing for the discrete linear system (3.2) when the following equation, referred to as the discrete-time algebraic Riccati equation

$$\boldsymbol{A}_{\boldsymbol{K}}^T \boldsymbol{P} \boldsymbol{A}_{\boldsymbol{K}} - \boldsymbol{P} = -\tilde{\boldsymbol{Q}} - \boldsymbol{K}^T \tilde{\boldsymbol{R}} \boldsymbol{K}, \tag{3.4}$$

$$\boldsymbol{K} = (\tilde{\boldsymbol{R}} + \boldsymbol{B}^T \boldsymbol{P} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P} \boldsymbol{A}, \tag{3.5}$$

accepts a symmetric positive-definite matrix $\boldsymbol{P}$. The objective function of this problem is

$$J = \left\|(\boldsymbol{x}(k) - \boldsymbol{x}_{sp})\right\|_{\boldsymbol{P}}^2, \tag{3.6}$$

which can be used as a first guess for an operating region.

Since this is the discrete-time case, *dlqr* or *dare* will compute the matrix $\boldsymbol{P}$ and the gain matrix $\boldsymbol{K}$. *dare* function does not allow for an input penalty matrix as input.

Another modification to the discretized model is the conversion from positional form, such as (3.2), to a velocity form such as

$$z(k+1) = \begin{bmatrix} \boldsymbol{x}(k+1) \\ \boldsymbol{u}(k) \end{bmatrix} = \boldsymbol{A}_v(\boldsymbol{z}(k) - z_{sp}) + \boldsymbol{B}_v \boldsymbol{\Delta u}(k), \tag{3.7}$$

where

$$\boldsymbol{A}_v := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ 0 & \boldsymbol{I} \end{bmatrix},$$
$$\boldsymbol{B}_v := \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{I} \end{bmatrix}.$$

By the extension of the state vector by the input vector, with the new input as the input move $\boldsymbol{\Delta u}$, the discrete-time model is now in velocity form, and one can proceed with bounds on both input and input moves. The matrices $\boldsymbol{A}_v$ and $\boldsymbol{B}_v$ can substitute the positional form state space model matrices ($\boldsymbol{A}$, $\boldsymbol{B}$) in DARE (3.4), while the objective function is changed to

$$J = \left\| (\boldsymbol{z}(k) - z_{sp}) \right\|_{\boldsymbol{P}}^2. \tag{3.8}$$

3.1 Semi-definite programming approach

Computing the operating region of a LQR subject to input and state bounds is not a novel topic. It is proposed by Sznaier & Damborg (1987) a constrained LQR in a constraint set described by a convex polyhedron, which is often how state and input bounds are described. The regulator control actions are computed via solution of an online quadratic programming (QP) problem. Then, Scokaert & Rawlings (1998) discusses this implementation also while discussing an important point regarding finiteness of QP problem to be solved on-line. However, there is no indication as to how to implement or solve the optimization problems proposed by Scokaert & Rawlings (1998).

Say that one desires to determine an operating region Ω where a state constraint such as

$$\mathscr{X} = \{\boldsymbol{x}(k) \in \mathbb{R}^{n_x} | \boldsymbol{x}_{min} \leq \boldsymbol{x}(k) \leq \boldsymbol{x}_{max}, k \geq 0\}. \tag{3.9}$$

must be satisfied. This set can be expressed as a set of inequalities such as

$$\begin{bmatrix} \boldsymbol{x}(k) \\ -\boldsymbol{x}(k) \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{x}_{max} \\ -\boldsymbol{x}_{min} \end{bmatrix},$$

which when rearrange, yields an inequality of the form $\boldsymbol{A}_c\boldsymbol{x} \leq \boldsymbol{b}$:

$$\begin{bmatrix} \boldsymbol{I}_{nx} \\ -\boldsymbol{I}_{nx} \end{bmatrix} \boldsymbol{x}(k) \leq \begin{bmatrix} \boldsymbol{x}_{max} \\ -\boldsymbol{x}_{min} \end{bmatrix}. \tag{3.10}$$

This can also be done for input constraints. In this case, the constraint (3.10) can have said constraints appended to it, such as

$$\begin{bmatrix} \boldsymbol{I}_{nx+nu} \\ -\boldsymbol{I}_{nx+nu} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{u}(k-1) \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{z}_{max} \\ -\boldsymbol{z}_{min} \end{bmatrix}. \tag{3.11}$$

Problem $\mathscr{P}_a$ is proposed as an attempt to produce an operating region where the closed-loop satisfies not only state and input constraints but input move constraints as well.

Problem $\mathscr{P}_a$:

$$\max_{r} \quad r, \tag{3.12}$$

subject to

$$\boldsymbol{A}_c r(\boldsymbol{z}(k) - \boldsymbol{z}_{sp}) \leq \boldsymbol{b}, \tag{3.12a}$$

$$\left|\left|(\boldsymbol{z}(k) - \boldsymbol{z}_{sp})\right|\right|^2_{\boldsymbol{P}} \leq 1, \tag{3.12b}$$

$$-\boldsymbol{\Delta u}_{max} \leq -\boldsymbol{K}(\boldsymbol{z}(k) - \boldsymbol{z}_{sp}) \leq \boldsymbol{\Delta u}_{max}. \tag{3.12c}$$

Where r is the radius of the operating region, computed as an ellipsoid with weighting matrix P. The radius is a decision variable for the SDP problem which must satisfy the extended state constraint (3.12a) as well as the input move constraint (3.12c).

3.2 LMI-based approach

Obtaining the regulator and operating region are much more convenient to be obtained through an LMI problem, since it can account for input, input move, and state constraints while yielding the largest feasible operating region with one single optimization problem. Through the LMI problem, the decision variables are the regulator and the terminal penalty matrix, while in the successive optimization problems - the operating region radius is the only decision variable. The LMI is capable of delivering larger operating regions at the expense of the regulator's gain.

Problem $\mathscr{P}_c$ is the corresponding LMI problem for computing penalty and gain matrices.

Problem $\mathscr{P}_c$:

$$\min_{\boldsymbol{W}_1,\boldsymbol{W}_2} -\log\det(\boldsymbol{W}_1)^{-1} \tag{3.13}$$

subject to

$$\boldsymbol{W}_1 > 0, \tag{3.13a}$$

$$\begin{bmatrix} -\boldsymbol{W}_1 & \boldsymbol{A}_v\boldsymbol{W}_1+\boldsymbol{B}_v\boldsymbol{W}_2 \\ \boldsymbol{W}_1\boldsymbol{A}_v^T+\boldsymbol{W}_2^T\boldsymbol{B}_v^T & -\boldsymbol{W}_1 \end{bmatrix} \leq 0 \tag{3.13b}$$

$$\begin{bmatrix} 1 & \begin{bmatrix} c_i'\boldsymbol{W}_1+d_i'\boldsymbol{W}_2 \end{bmatrix} \\ * & \boldsymbol{W}_1 \end{bmatrix} \geq 0, i=1,\ldots,n_i. \tag{3.13c}$$

with

$$\boldsymbol{P}=\boldsymbol{W}_1^{-1} \tag{3.14}$$

$$\boldsymbol{K}=\boldsymbol{W}_2\boldsymbol{W}_1^{-1}. \tag{3.15}$$

The stability constraint (3.13b) was shown in Duan & Yu (2013), Theorem 6.7, which says that the discrete-time linear system to be controlled is stabilizable if and only if there exist a symmetric matrix $\boldsymbol{W}_1$ and a matrix $\boldsymbol{W}_2$ satisfying the LMI (3.13b). It can be proved by way of the LMI (3.16), which guarantees that the linear system in question is stabilizable if and only if there exist a matrix $\boldsymbol{K}$ and a symmetric matrix $\boldsymbol{P}$ that satisfy

$$\begin{bmatrix} -\boldsymbol{P} & (\boldsymbol{A}_v-\boldsymbol{B}_v\boldsymbol{K})\boldsymbol{P} \\ \boldsymbol{P}(\boldsymbol{A}_v-\boldsymbol{B}_v\boldsymbol{K})^T & -\boldsymbol{P} \end{bmatrix} \leq 0. \tag{3.16}$$

Having $\boldsymbol{W}_2=\boldsymbol{KP}$ and substituting it in the expression above, we obtain the stability constraint (3.13b). If the matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ exist, then by expression (3.15) as well we arrive at the condition (3.16).

The state and input constraint (3.13c) represents an inscribed ellipsoid to the constraint set

$$\mathscr{Z}_c := \left\{ |\boldsymbol{c}_i'+\boldsymbol{d}_i'\boldsymbol{K}|(\boldsymbol{z}(k)-\boldsymbol{z}_{sp}) \leq 1 \right\}. \tag{3.17}$$

This set is a constraint of the form $\boldsymbol{A}_c\boldsymbol{x} \leq \boldsymbol{b}$, with the RHS scaled to unit value.

The demonstration of the effect of the constraint (3.13c) begins with its manipulation. Due to positive definiteness of $\boldsymbol{W}_1$, as per constraint (3.13a), we can rewrite this expression as:

$$1+(\boldsymbol{c}_i'\boldsymbol{W}_1+\boldsymbol{d}_i'\boldsymbol{W}_2)(\boldsymbol{W}_1)^{-1}(\boldsymbol{c}_i'\boldsymbol{W}_1+\boldsymbol{d}_i'\boldsymbol{W}_2)^T \geq 0, i=1,\ldots,n_i \tag{3.18}$$

where n_i represents the number of inequalities used to define $\mathscr{Z}_c$.

We then take the Schur complement of (3.18). Schur complements enables establishing the equivalence between the following expressions:

$$\boldsymbol{Q} - \boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}^T > 0,$$
$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & \boldsymbol{R} \end{bmatrix} > 0.$$

Schur complement of expression (3.18) is, after substitutions of expressions (3.14) and (3.15) and remembering once again the positive definiteness of $\boldsymbol{W}_1$ as per constraint (3.13a):

$$1 - (\boldsymbol{c}_i' + \boldsymbol{d}_i'\boldsymbol{K})\boldsymbol{P}(\boldsymbol{c}_i' + \boldsymbol{d}_i'\boldsymbol{K})^T \geq 0,$$

which is the inscribed ellipsoid mentioned earlier.

This is not the method proposed by Rajhans et al. (2019), where it is suggested that the practictioner obtain the regulator through Matlab's function *dlqr*, then successive optimizations are to be solved in order to determine the ellipsoid radius γ where

$$\Omega_\gamma = \left\{ \boldsymbol{z}(k) \in \mathscr{X} \times \mathscr{U} : \left\| \boldsymbol{z}(k) - \boldsymbol{z}_{sp} \right\|_{\boldsymbol{P}}^2 \leq \gamma, -\boldsymbol{K}(\boldsymbol{z}(k) - \boldsymbol{z}_{sp}) \in \mathscr{U}_\Delta \right\}. \tag{3.19}$$

Lemma 1 states when a discrete-time regulator stabilizes a discrete-time nonlinear system.

Lemma 1 (adapted from Lemma 8 of Rajhans et al. (2019)) *A discrete-time nonlinear system* (3.1) *controlled by the LQR that solves the DAREs*

$$\boldsymbol{A}_{\boldsymbol{K}}^T \boldsymbol{P} \boldsymbol{A}_{\boldsymbol{K}} - \boldsymbol{P} = -\tilde{\boldsymbol{Q}} - \boldsymbol{K}^T \tilde{\boldsymbol{R}} \boldsymbol{K} \tag{3.20}$$
$$\boldsymbol{K} = (\tilde{\boldsymbol{R}} + \boldsymbol{B}_v^T \boldsymbol{P} \boldsymbol{B}_v)^{-1} \boldsymbol{B}_v^T \boldsymbol{P} \boldsymbol{A}_v \tag{3.21}$$

is asymptotically stable for all states inside the operating region (3.19) *if the nonlinearity* $\varphi(\boldsymbol{z}(k))$

$$\varphi(\boldsymbol{z}(k)) = (\boldsymbol{z}(k) - \boldsymbol{z}_{sp})^T \boldsymbol{\Delta Q}_d (\boldsymbol{z}(k) - \boldsymbol{z}_{sp}) - 2\boldsymbol{\Psi}(\boldsymbol{z}(k))^T \boldsymbol{P}\boldsymbol{A}_{\boldsymbol{K}} (\boldsymbol{z}(k) - \boldsymbol{z}_{sp}) - \boldsymbol{\Psi}(\boldsymbol{z}(k))^T \boldsymbol{P} \boldsymbol{\Psi}(\boldsymbol{z}(k)) \tag{3.22}$$

is non-negative.

We have $\boldsymbol{P}$ and $\boldsymbol{K}$ as the unique positive-definite solution for the DARE, penalty matrix and gain matrix respectively. As the DARE admits a solution, the closed-loop represented by the matrix $\boldsymbol{A}_{\boldsymbol{K}} = \boldsymbol{A}_v - \boldsymbol{B}_v \boldsymbol{K}$ is Hurwitz, that is, asymptotically stable. Rewriting the discrete nonlinear system in terms of the closed-loop matrix, which is where $\boldsymbol{\Psi}(\boldsymbol{z}(k))$ is shown, we have:

$$z(k+1) = \boldsymbol{A_K}(z(k) - z_{sp}) + \boldsymbol{\Psi}(z(k)) \tag{3.23}$$

$$\boldsymbol{\Psi}(z(k)) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k-1) - \boldsymbol{K}(z(k) - z_{sp})) - \boldsymbol{A_K}(z(k) - z_{sp}). \tag{3.24}$$

Consider now a candidate Lyapunov function for the nonlinear system controlled by the LQR $\boldsymbol{K}$ inside the operating region Ω:

$$V_t(z(k)) = \left\|z(k) - z_{sp}\right\|_{\boldsymbol{P}}^2$$

Comparing the candidate Lyapunov functions for times k and $k+1$:

$$\begin{aligned} V_t(z(k+1)) - V_t(z(k)) = {} & (z(k) - z_{sp})^T (\boldsymbol{A_K^T P A_K} - \boldsymbol{P})(z(k) - z_{sp}) \\ & + 2\boldsymbol{\Psi}(z(k))^T \boldsymbol{P A_K}(z(k) - z_{sp}) + \boldsymbol{\Psi}(z(k))^T \boldsymbol{P \Psi}(z(k)) \end{aligned} \tag{3.25}$$

Defining the RHS of the DARE (3.4) as

$$\boldsymbol{Q}_d := -\boldsymbol{Q}_x - \boldsymbol{K}^T \boldsymbol{R K}, \tag{3.26}$$

$$\boldsymbol{\Delta Q}_d := \boldsymbol{\Delta Q} + \boldsymbol{K}^T \boldsymbol{\Delta R K}, \tag{3.27}$$

$$\boldsymbol{\Delta Q} := \tilde{\boldsymbol{Q}} - \boldsymbol{Q} \geq \boldsymbol{0}, \tag{3.28}$$

$$\boldsymbol{\Delta R} := \tilde{\boldsymbol{R}} - \boldsymbol{R} \geq \boldsymbol{0} \tag{3.29}$$

noticing that the DARE was modified in order to accommodate the tuning parameters $\boldsymbol{\Delta Q}$ and $\boldsymbol{\Delta R}$, that enlarge the operating region:

$$\boldsymbol{A_K^T P A_K} - \boldsymbol{P} = -(\boldsymbol{Q}_d + \boldsymbol{\Delta Q}_d)$$

and substituting it in equation (3.25), we have

$$V_t(z(k+1)) - V_t(z(k)) = (z(k) - z_{sp})^T (\boldsymbol{Q}_d)(z(k) - z_{sp}) + \varphi(z(k)) \tag{3.30}$$

$$\varphi(z(k)) = \left\|z(k) - z_{sp}\right\|_{\boldsymbol{\Delta Q}_d}^2 - 2\boldsymbol{\Psi}(z(k))^T \boldsymbol{P A_K}(z(k) - z_{sp}) - \boldsymbol{\Psi}(z(k))^T \boldsymbol{P \Psi}(z(k)). \tag{3.31}$$

The induced norm of the operator $\boldsymbol{\Psi}(z(k))$ is defined as

$$\beta_\Psi = \sup \left\{ \frac{\|\boldsymbol{\Psi}(z(k))\|}{\left\|z(k) - z_{sp}\right\|} \middle| z(k) \in \Omega, z(k) \neq z_{sp} \right\} \tag{3.32}$$

which grants useful inequalities that help defining a bound for nonlinearity $\varphi(\boldsymbol{z}(k))$:

$$\boldsymbol{\Psi}(\boldsymbol{z}(k))^T \boldsymbol{P}\boldsymbol{A}_{\boldsymbol{K}}(\boldsymbol{z}(k)-\boldsymbol{z}_{sp}) \leq \beta_{\boldsymbol{\Psi}} \left\|\boldsymbol{P}\boldsymbol{A}_{\boldsymbol{K}}\right\| \left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|^2 \tag{3.33}$$

$$\boldsymbol{\Psi}(\boldsymbol{z}(k))^T \boldsymbol{P}\boldsymbol{\Psi}(\boldsymbol{z}(k)) \leq \beta_{\boldsymbol{\Psi}}^2 \left\|\boldsymbol{P}\right\| \left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|^2 \tag{3.34}$$

$$\left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|^2_{\boldsymbol{\Delta Q}_d} \geq \lambda_{min}(\boldsymbol{\Delta Q}_d) \left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|^2 \tag{3.35}$$

When combining inequalities 3.33 and 3.34 with the comparison 3.31, we have that the comparison has an upper bound:

$$\begin{aligned} V_t(\boldsymbol{z}(k+1)) - V_t(\boldsymbol{z}(k)) \leq & - \left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|^2_{\boldsymbol{Q}_d} \\ & -(\lambda_{min}(\boldsymbol{\Delta Q}_d) - 2\beta_{\boldsymbol{\Psi}} \left\|\boldsymbol{P}\boldsymbol{A}_{\boldsymbol{K}}\right\| - \beta_{\boldsymbol{\Psi}}^2 \left\|\boldsymbol{P}\right\|) \left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|^2 \end{aligned}$$

With this inequality, we have an expression that relates the non-increasing behavior of the objective function value $V_t(\boldsymbol{z}(k))$ with the induced norm of the nonlinearity $\boldsymbol{\Psi}(\boldsymbol{z}(k))$.

$$\lambda_{min}(\boldsymbol{\Delta Q}_d) - 2\beta_{\boldsymbol{\Psi}} \left\|\boldsymbol{P}\boldsymbol{A}_{\boldsymbol{K}}\right\| - \beta_{\boldsymbol{\Psi}}^2 \left\|\boldsymbol{P}\right\| = 0$$

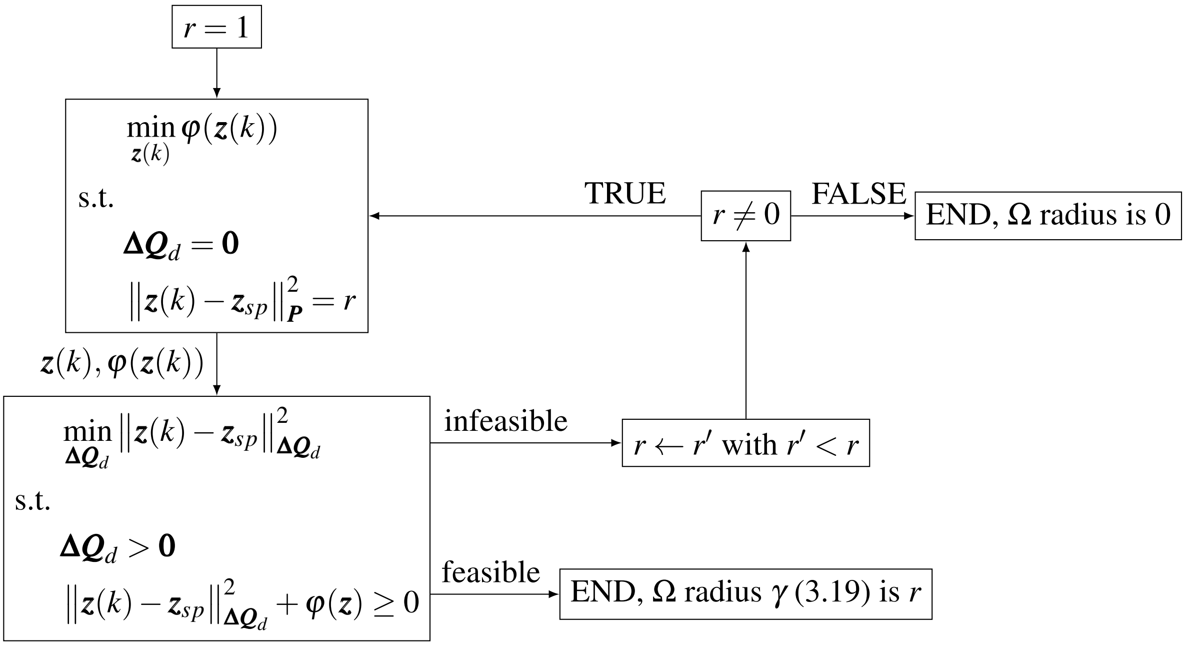
that is, an upper bound for the induced norm of the nonlinearity $\beta_{\boldsymbol{\Psi}}$:

$$\beta_{\boldsymbol{\Psi}}^* = \frac{-\left\|\boldsymbol{P}\boldsymbol{A}_{\boldsymbol{K}}\right\| + \sqrt{\left\|\boldsymbol{P}\boldsymbol{A}_{\boldsymbol{K}}\right\|^2 + \lambda_{min}(\boldsymbol{\Delta Q}_d)\left\|\boldsymbol{P}\right\|}}{\left\|\boldsymbol{P}\right\|}$$

The induced norm $\beta_{\boldsymbol{\Psi}}$ along with its upper bound $\beta_{\boldsymbol{\Psi}}^*$ can be used to determine where in the terminal region Ω the inequality $\varphi(\boldsymbol{z}(k)) \geq 0$ is satisfied, indicating asymptotic stability of the nonlinear system when controlled by the linear quadratic regulator computed by DAREs (3.20) and (3.21). ■

With this theoretical result, one has a direct expression that indicates if a closed-loop system composed of a LQR and a nonlinear system is asymptotically stable. For the case where the regulator is computed through the LMIP $\mathscr{P}_c$, the existence of a positive-definite matrix $\boldsymbol{\Delta Q}_d$, large enough in order to make the nonlinearity $\boldsymbol{\varphi}(\boldsymbol{z})$ be non-negative, is checked through two optimization problems: one that computes the value of the nonlinearity for a null $\boldsymbol{\Delta Q}_d$ at the border of the operating region, and another that computes a positive-definite $\boldsymbol{\Delta Q}_d$ matrix that makes the nonlinearity value be zero. The following algorithm demonstrates this procedure.

Figure 10 – Algorithm for nonlinearity bound identification of closed-loop with nonlinear system and LQR


$r = 1$
$\min_{\boldsymbol{z}(k)} \varphi(\boldsymbol{z}(k))$
s.t.
$\boldsymbol{\Delta Q}_d = \boldsymbol{0}$
$\|\boldsymbol{z}(k) - \boldsymbol{z}_{sp}\|^2_{\boldsymbol{P}} = r$
TRUE
$r \neq 0$
FALSE
END, Ω radius is 0
$\boldsymbol{z}(k), \varphi(\boldsymbol{z}(k))$
$\min_{\boldsymbol{\Delta Q}_d} \|\boldsymbol{z}(k) - \boldsymbol{z}_{sp}\|^2_{\boldsymbol{\Delta Q}_d}$
s.t.
$\boldsymbol{\Delta Q}_d > \boldsymbol{0}$
$\|\boldsymbol{z}(k) - \boldsymbol{z}_{sp}\|^2_{\boldsymbol{\Delta Q}_d} + \varphi(\boldsymbol{z}) \geq 0$
infeasible
$r \leftarrow r'$ with $r' < r$
feasible
END, Ω radius γ (3.19) is r

Source: Own author.

This algorithm was developed as an alternative to the proposal of Rajhans et al. (2019), which consists of tuning the original DARE equation with increasingly larger state and input penalty matrices $\tilde{\boldsymbol{Q}}$ and $\tilde{\boldsymbol{R}}$ in order to achieve a $\boldsymbol{\Delta Q}_d$ matrix sufficiently high that it is capable of making the nonlinearity $\varphi(\boldsymbol{z})$ nonpositive. This trial and error method is replaced by two optimization problems. They are proposed in order to compute such $\boldsymbol{\Delta Q}_d$ matrix for the radius of the operating region where the constrained LQR acts. When such matrix does not exist, with the second optimization problem infeasible, the algorithm proposes to diminish the guess radius r and the optimization problems are to be solved again, until the optimization problems are feasible or no $\boldsymbol{\Delta Q}_d$ is found. When it is not found, it can be said that the terminal region consists of the reference value of the regulator.

If the practitioner desires, the matrix $\boldsymbol{\Delta Q}_d$ can be used to determine a possible tuning of $\boldsymbol{Q}_x$ and $\boldsymbol{R}$ that satisfies the redefinition of the DARE equation with the expressions (3.26)-(3.29).

3.3 Case study: quadruple tank system controlled by LQR in discrete-time

In this section, the system to be controlled by an LQR is first exposed. Its first principle model and physical limitations are exposed. Then, the operating regions computed by the methods shown in the previous sections are compared visually. Finally, the controllers computed by both methods (Problems $\mathscr{P}_a$ and $\mathscr{P}_c$) are simulated in closed-loop with the nonlinear system in question, with exposition of results obtained.

3.3.1 Quadruple tank system

A common benchmark to control systems is the quadruple-tank system, which describes the liquid level of an association of tanks in parallel and series that are fed by two pumps. The ODE system that describes this system states over time is as follows:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -\frac{a_1}{A_1}\sqrt{2gx_1} + \frac{a_3}{A_1}\sqrt{2gx_3} \\ -\frac{a_2}{A_2}\sqrt{2gx_2} + \frac{a_4}{A_2}\sqrt{2gx_4} \\ -\frac{a_3}{A_3}\sqrt{2gx_3} \\ -\frac{a_4}{A_4}\sqrt{2gx_4} \end{bmatrix} + \begin{bmatrix} \frac{\gamma_1}{A_1} \\ 0 \\ 0 \\ \frac{1-\gamma_1}{A_4} \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ \frac{\gamma_2}{A_2} \\ \frac{1-\gamma_2}{A_3} \\ 0 \end{bmatrix} u_2, \tag{3.36}$$

where x_i represents the i-th tank level [cm], u_i represents the i-th pump flowrate [ml.s^{-1}], a_i representing the i-th tank's outlet transversal area [cm^2], A_i representing the i-th tank's transversal area [cm^2], γ_i representing the i-th pump's split ratio, and g as the gravitational acceleration [m.s^{-2}]. The numerical values of these constants were estimated by Raff et al. (2006) in its experimental module - and to be used in this work - are:

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} 50.27 & 50.27 & 28.27 & 28.27 \end{bmatrix}, \\ \boldsymbol{a} &= \begin{bmatrix} 0.233 & 0.242 & 0.127 & 0.127 \end{bmatrix}, \\ g &= 9.81,\ \boldsymbol{\gamma} = \begin{bmatrix} 0.4 & 0.4 \end{bmatrix}. \end{aligned}$$

Figure 11 represents the diagram of the quadruple tank system used in this work.

Limitations of the mathematical model (based on Bernoulli's law for the tank emptying term) were adopted as hard state constraints, represented by set $\mathscr{X}$. Physical limitations on the pump's minimum and maximum flowrate were adopted as hard input constraints, represented by set $\mathscr{U}$. As an input move constraint, it was considered that the pumps cannot change flowrate faster than 1 ml.s^{-2}. With sampling of 4s, this corresponds to a limitation to the input move of 4ml.s^{-1}, which is then represented by set $\mathscr{U}_\Delta$:

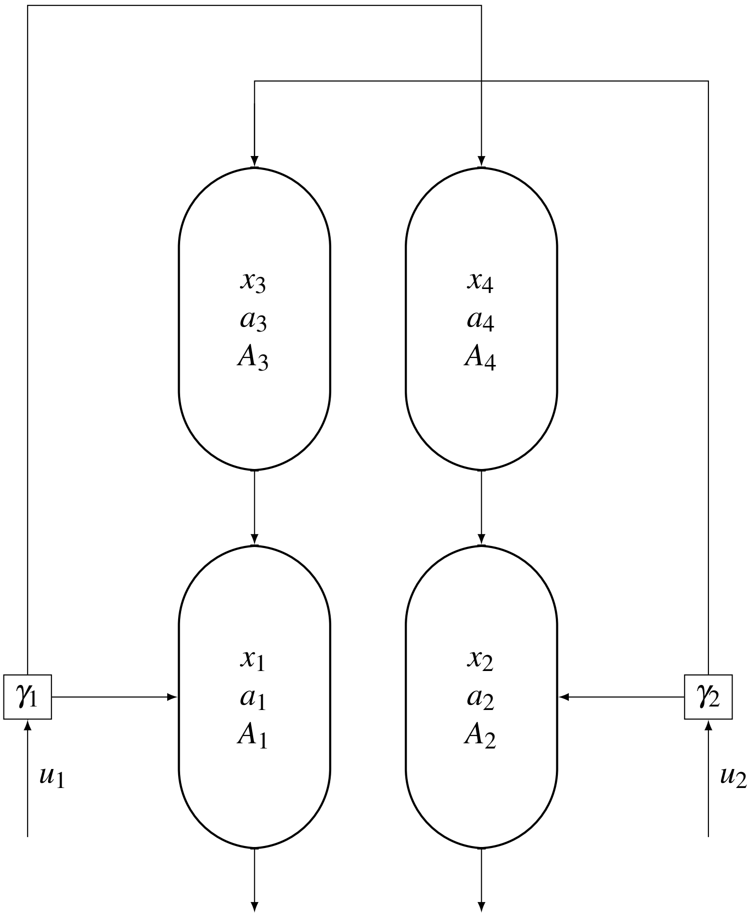
$$\begin{aligned} \mathscr{X} &= \left\{ \boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix} 7.5 & 7.5 & 3.5 & 4.5 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 28 & 28 & 28 & 28 \end{bmatrix}^T \right\}, \\ \mathscr{U} &= \left\{ \boldsymbol{u} \in \mathbb{R}^{n_u} \middle| \begin{bmatrix} 0 & 0 \end{bmatrix}^T \leq \boldsymbol{u} \leq \begin{bmatrix} 60 & 60 \end{bmatrix}^T \right\}, \\ \mathscr{U}_\Delta &= \left\{ \boldsymbol{\Delta u} \in \mathbb{R}^{n_u} \middle| \begin{bmatrix} -4 & -4 \end{bmatrix}^T \leq \boldsymbol{\Delta u} \leq \begin{bmatrix} 4 & 4 \end{bmatrix}^T \right\}. \end{aligned}$$

3.3.2 Operating region and LQR computation

The methods exposed in the previous sections of this chapter were implemented in MATLAB (THE MATHWORKS, INC., 2017). Problems $\mathscr{P}_a$ and $\mathscr{P}_c$ were written in YALMIP (LÖFBERG, 2004). The first was solved by algorithm SeDuMi-1.3 (STURM, 1999) while the latter was solved via SDPT3 (TÜTÜNCÜ; TOH; TODD, 2003). The first optimization problem in the algorithm shown in Figure 10 was written in CasADi with IPOpt as its solver, while the second was written in YALMIP with SDPT3 as its solver.

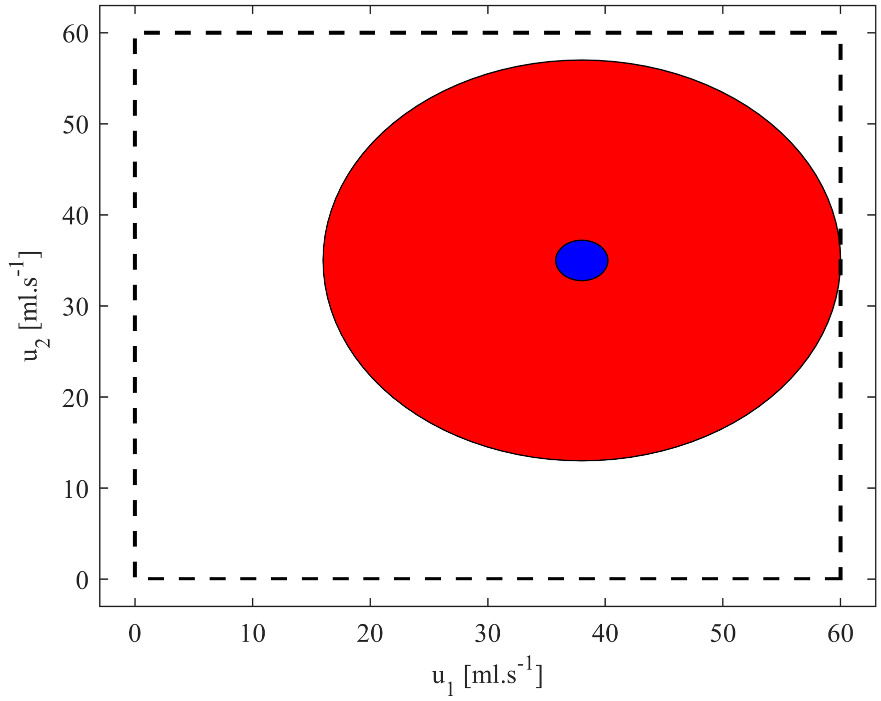
The resulting operating regions are seen in Figures 12 and 13. In the state space, the proposed region computed via LMIP presents significantly larger operating terminal region, when compared to the SDP method of computing penalty matrix $\boldsymbol{P}$ and gain $\boldsymbol{K}$ through DARE then identifying a radius where state, input, and input move bounds are satisfied. The difference in the operating region in the input space is closely related to the controller gain of each formulation. The resulting region from the LMIP is larger than the SDP method due to the controller gain being one of the decision variables when it computes the terminal region. When the regulator gain is a decision variable, the solution of the optimization will consist of smaller gains in order

Figure 11 – Quadruple-tank system diagram



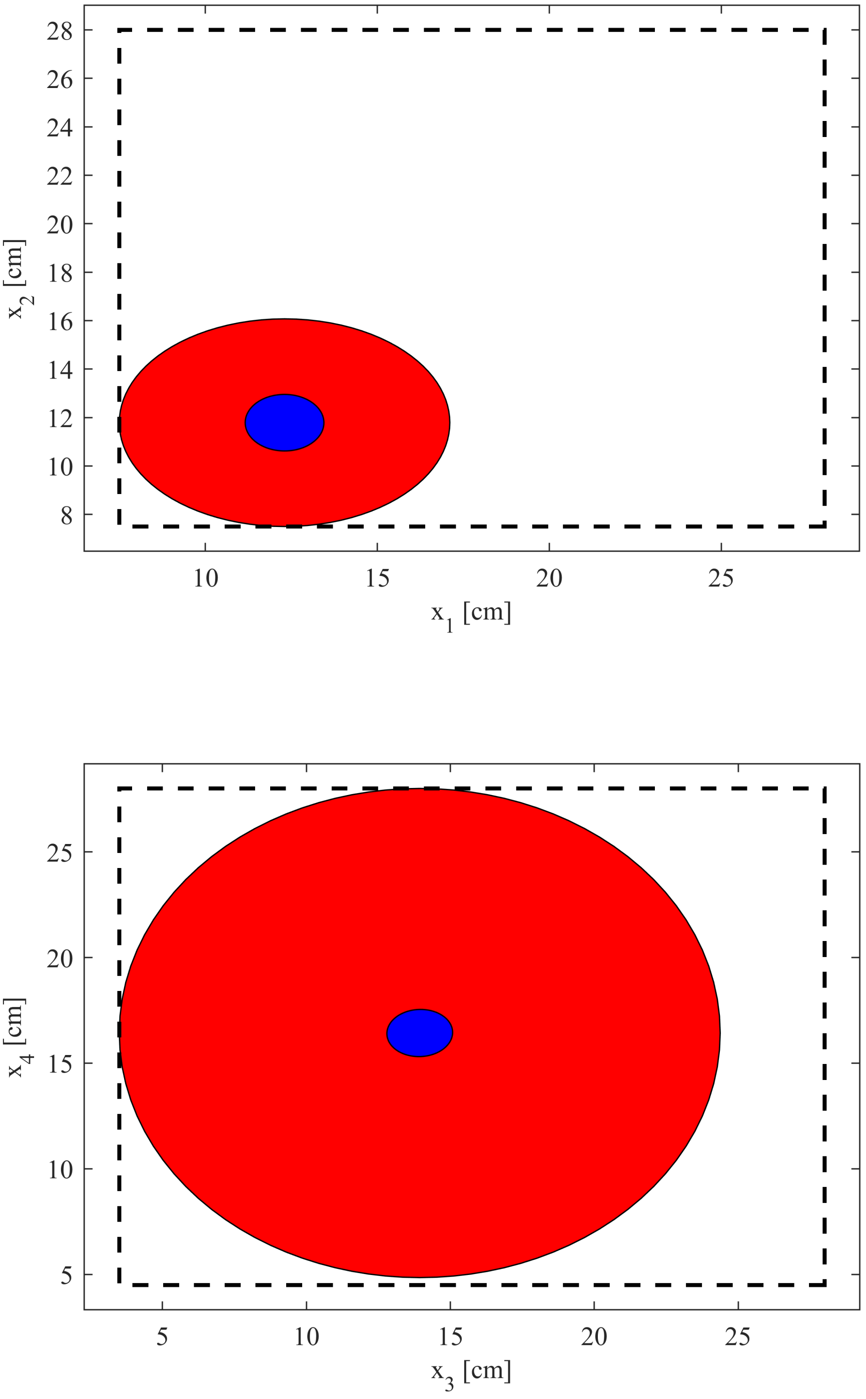
Source: Own author

Figure 12 – LQR operating region in the input space (- - -) for Problem $\mathscr{P}_a$ (●) and Problem $\mathscr{P}_c$ (●)



to enlarge the operating region radius, which is to be maximized - as per objective function of Problem $\mathscr{P}_c$.

Figure 13 – LQR operating region in the state space (- - -) for Problem $\mathscr{P}_a$ (●) and Problem $\mathscr{P}_c$ (●)



The penalty matrix $\boldsymbol{P}$, controller gain $\boldsymbol{K}$, and radius r are

$$\boldsymbol{P}_c = \begin{bmatrix} 43.3510 & 0 & 0 & 0 & 0 & 0 \\ 0 & 54.4376 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9.1822 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.4666 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.0661 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.0661 \end{bmatrix} \times 10^{-3}, \tag{3.37}$$

$$\boldsymbol{K}_c = \begin{bmatrix} 0.1936 & 0.0813 & 0.0257 & 0.1167 & 0.1126 & 0.0007 \\ 0.0798 & 0.2165 & 0.1150 & 0.0395 & 0.0018 & 0.1175 \end{bmatrix}, \tag{3.38}$$

$$r_c = 1. \tag{3.39}$$

Considering the operating region has a fixed radius of 1, the LMIP was capable of maximizing the operating region under such constraint, by determining smaller values of the penalty matrix while varying controller gain. The resulting regulator gain and penalty matrix for Problem $\mathscr{P}_a$, now in discrete-time, are

$$\boldsymbol{P}_a = \begin{bmatrix} 6.3856 & -0.1410 & 1.2763 & -0.4704 & 0.0790 & 0.0560 \\ -0.1410 & 6.0693 & -0.4140 & 1.140 & 0.0482 & 0.0759 \\ 1.2763 & -0.4140 & 6.4555 & -0.3262 & 0.0095 & 0.2698 \\ -0.4704 & 1.140 & -0.3262 & 6.6913 & 0.2806 & 0.0063 \\ 0.0790 & 0.0482 & 0.0095 & 0.2806 & 1.6343 & 0.0014 \\ 0.0560 & 0.0759 & 0.2698 & -0019 & 0.0014 & 1.6337 \end{bmatrix} \tag{3.40}$$

$$\boldsymbol{K}_a = \begin{bmatrix} 0.0790 & 0.0482 & 0.0095 & 0.2806 & 0.6343 & 0.0014 \\ 0.0560 & 0.0759 & 0.2698 & 0.0063 & 0.0014 & 0.6337 \end{bmatrix} \tag{3.41}$$

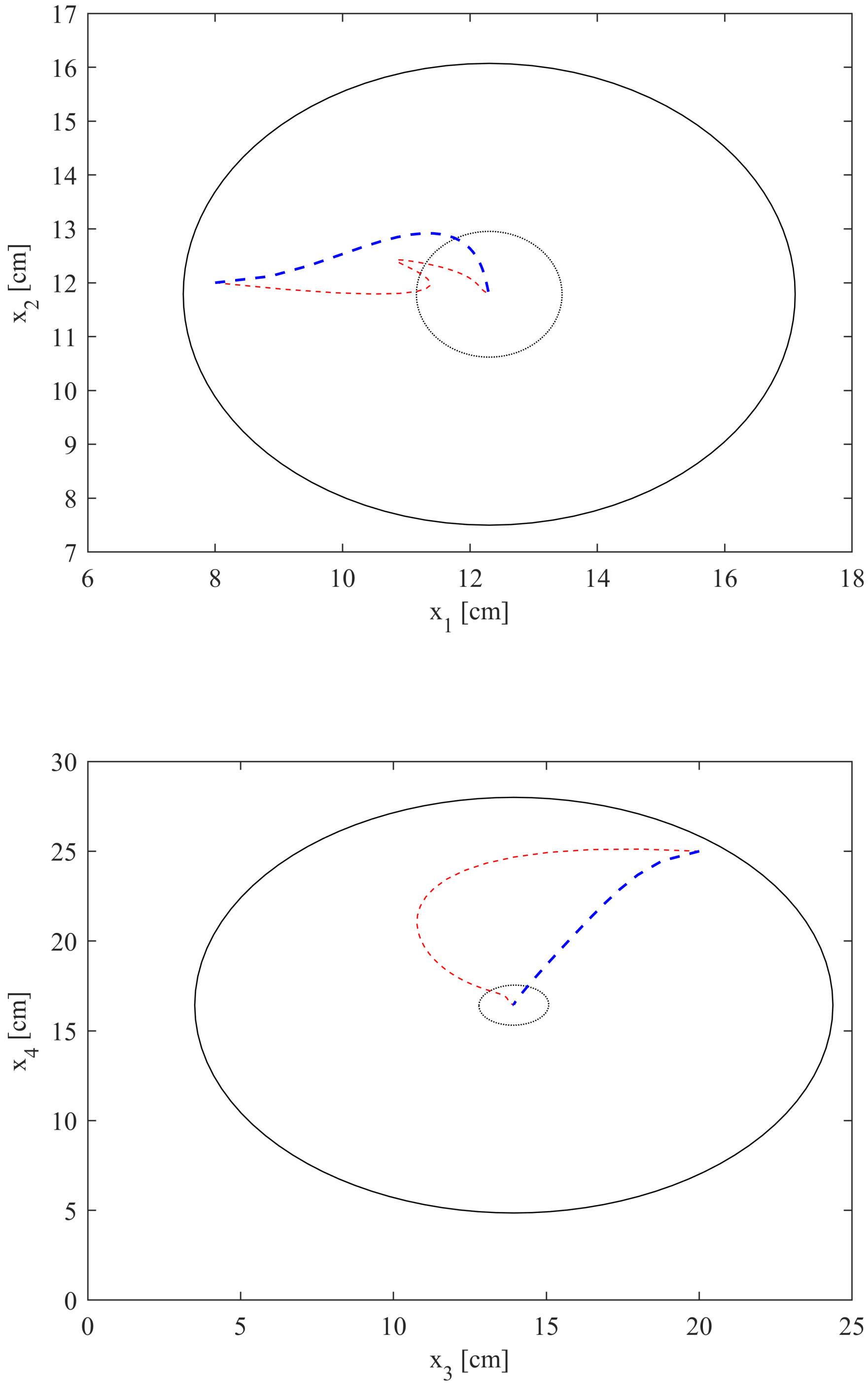
$$r_a = 7.9813 \tag{3.42}$$

The radii and penalty matrices are not directly comparable, however, the figures just exposed demonstrate that the LMIP approach produces larger operating regions. The region computed via LMIP $\mathscr{P}_c$ is larger due to the capability of the optimization problem of changing both the regulator gain $\boldsymbol{K}$ and the objective funciton penalty matrix $\boldsymbol{P}$. The SDP method is unable to change these values, hence its optimization problem will only compute the radius of the operating region where the state and input constraints are satisfied.

3.3.3 Closed-loop simulation

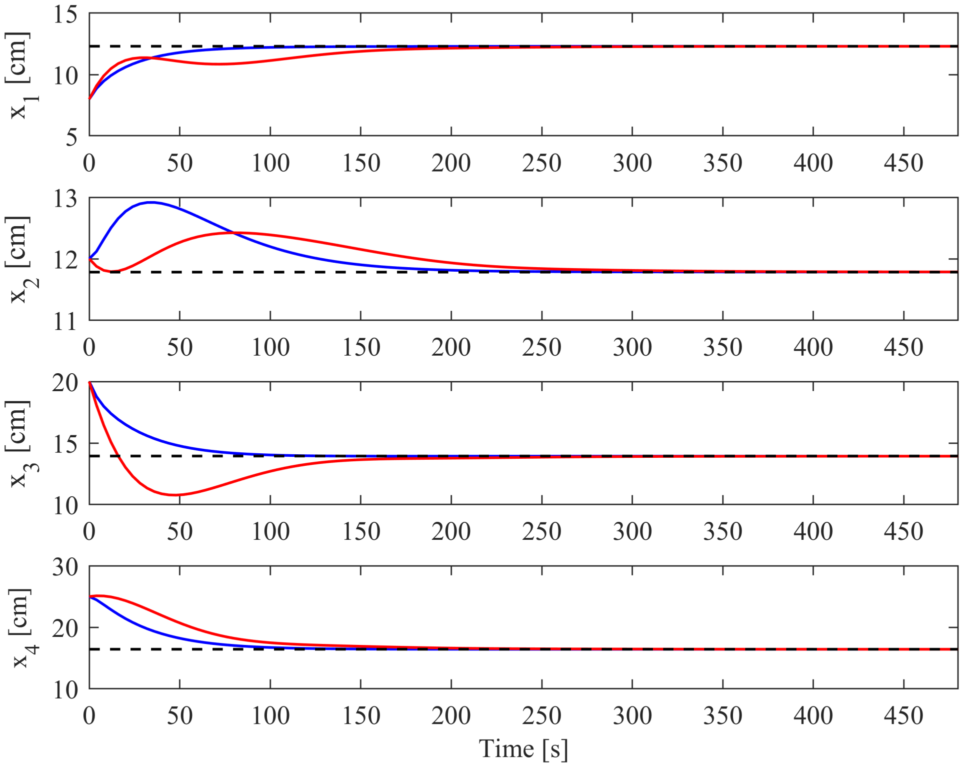
The regulator computed via $\mathscr{P}_c$ was simulated in closed-loop with the nonlinear system under study. Figure 14 shows states in pairs.

Figure 14 – Closed-loop under controllers computed via $\mathscr{P}_a$ and $\mathscr{P}_c$ with operating regions computed via $\mathscr{P}_a$ (○) and $\mathscr{P}_c$ (◌): state trajectory regulated by $\mathscr{P}_a$ (- - -) and $\mathscr{P}_c$ (- - -)



The regulators successfully bring the states towards its operating region center, the reference. The controller obtained via SDP Problem $\mathscr{P}_a$ shows better performance, as was expected, since the LMI method prioritizes the operating region size over the regulator gain magnitude. The state evolution over time is seen in Figure 15.

Figure 15 – Reference (- - -) and state profile over time under controllers computed via $\mathscr{P}_a$ (—) and $\mathscr{P}_c$ (—)



System states in closed-loop show oscillatory behavior. Drifting states were observed in Raff et al. (2006) in the case of closed-loop with a NMPC without stability constraints. In said work, the implemented formulation had no guaranteed stability, hence the states in closed-loop would drift until reaching a state boundary, where it would stop.

This regulator has guaranteed closed-loop stability with a nonlinear system according to the algorithm in Figure 10. Before that algorithm is executed, the eigenvalues of the Jacobian linearization in closed-loop with the computed gain $\boldsymbol{K}_c$, the controller which shows significantly more oscillatory behavior, are obtained and seen in Figure 16.

All the eigenvalues lie within the unit circle, indicating stability of the autonomous system $\boldsymbol{A}_v - \boldsymbol{B}_v\boldsymbol{K}_c$. Since the system presents eigenvalues with imaginary parts, oscillatory behavior is expected. However, this is not enough for stating stability for the nonlinear system.

The computed inputs are displayed in Figures 17 and 18.

Figure 16 – Eigenvalues of closed-loop system with controller computed by $\mathscr{P}_c$ within the unit circle. Marker × indicate eigenvalues

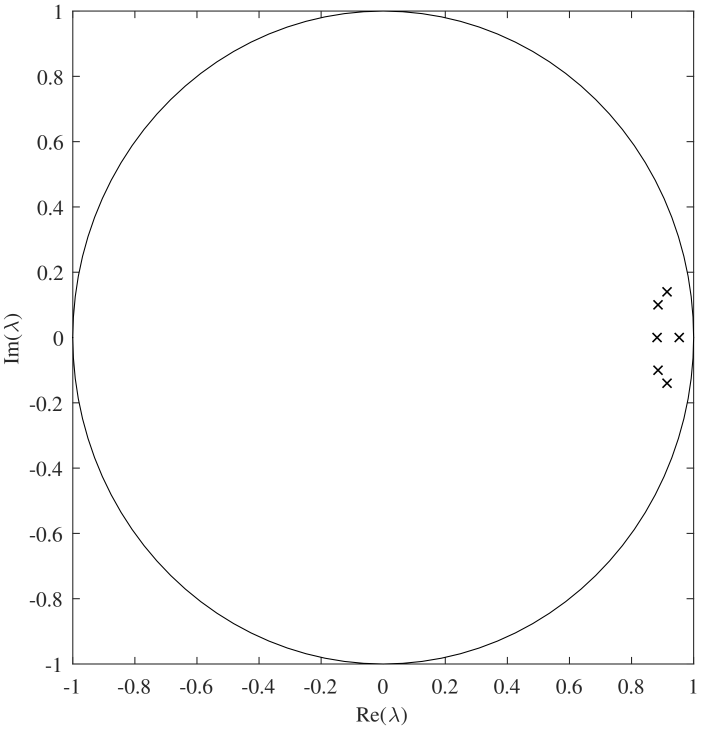
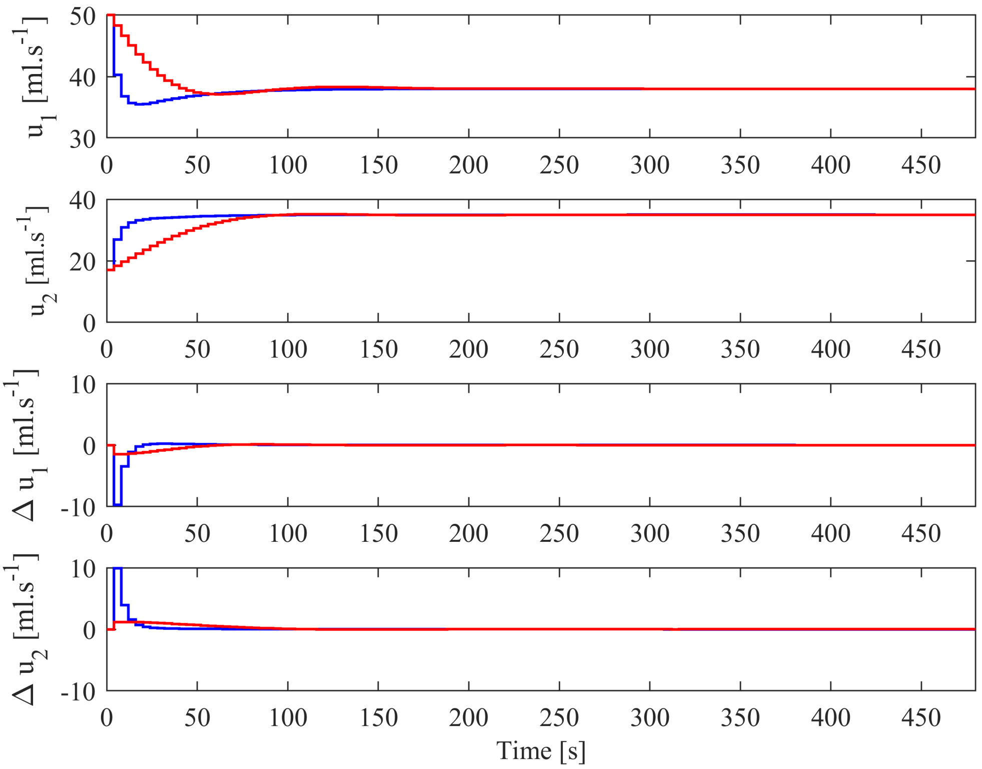
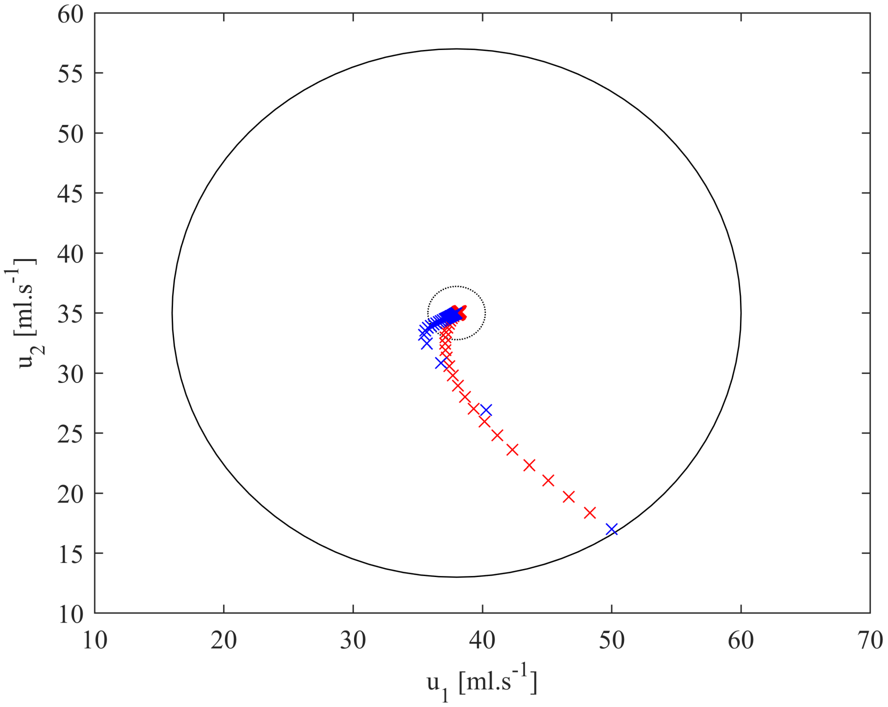


Figure 17 – Input profile over time determined by controller computed via $\mathscr{P}_c$



Inputs also converge to the reference value. However, the SDP-based controller shows

Figure 18 – Input evolution of controllers computed via $\mathscr{P}_a$ (×) and $\mathscr{P}_c$ (×) with operating regions computed via $\mathscr{P}_a$ (◌) and $\mathscr{P}_c$ (○)



constraint violation as it takes input moves larger than allowed. This is due to operation outside its constrained operating region, not shown here but it is known by way of Figures 13 and 12 that its operating region is much smaller. As a result, this controller will violate the input move constraints. Input evolution over time can be seen in Figure 17.

All input and input move constraints are satisfied in this simulation for the LMI-based controller Problem $\mathscr{P}_c$. The SDP-based controller does violate the input move constraints at the start of the simulation, due to its significantly larger gain.

Finally, $\boldsymbol{\Delta Q}$ matrices for both formulations that indicate the existence of the nonlinearity bound, computed as per algorithm of Figure 10, are

$$\boldsymbol{\Delta Q}_{d,a} = \begin{bmatrix} 10.1583 & 0.0004 & 0.0013 & -0.0010 & 0.0266 & -0.0365 \\ 0.0004 & 10.1578 & -0.0024 & 0.0018 & -0.0493 & 0.0674 \\ 0.0013 & -0.0024 & 10.1502 & 0.0062 & -0.1717 & 0.2346 \\ -0.0010 & 0.0018 & 0.0062 & 10.1539 & 0.1276 & -0.1751 \\ 0.0266 & -0.0493 & -0.1717 & 0.1276 & 6.6272 & 4.8258 \\ -0.0365 & 0.0674 & 0.2346 & -0.1751 & 4.8258 & 3.5359 \end{bmatrix},$$

$$\boldsymbol{\Delta Q}_{d,c} = \begin{bmatrix} 10.5397 & 0.0001 & 0.0007 & -0.0011 & 0.0201 & -0.0308 \\ 0.0001 & 10.5397 & -0.0008 & 0.0010 & -0.0231 & 0.0289 \\ 0.0007 & -0.0008 & 10.5350 & 0.0058 & -0.1350 & 0.1686 \\ -0.0011 & 0.0010 & 0.0058 & 10.5314 & 0.1615 & -0.2472 \\ 0.0201 & -0.0231 & -0.1350 & 0.1615 & 6.7592 & 4.7199 \\ -0.0308 & 0.0289 & 0.1686 & -0.2472 & 4.7199 & 3.3152 \end{bmatrix}.$$

When these matrices exist and are determined, it can be affirmed that the nonlinear system is closed-loop asymptotically stable when inside the operating region Ω. These matrices were determined without any further reductions of the operating region from their respective constrained operating region radii.

Part III

Nonlinear model predictive control

4 Nonlinear model predictive control formulations

Nonlinear model predictive control consists of computing a control policy for a plant through online solution of an optimal control problem. These controllers are formulated via an optimal control problem. This optimal control problem consists of predicting states through a nonlinear model of said plant in order to achieve an objective such as tracking a set point, minimizing input use or other objective.

First, an optimal control problem is shown. Then, it is shown how it is used in representing the nonlinear model predictive controller. Then, some strategies regarding stability of nonlinear model predictive controllers will be exposed and finally results regarding implementation of nonlinear model predictive controllers.

4.1 Optimal control theory

An optimal control problem consists of computing control actions to be injected into a process in order to obtain an optimal trajectory regarding the objective of the controller. These objectives are used to classify the optimal control problems into categories regarding the fixed or free end points and final times.

The optimization problem $\mathscr{P}_1$ represents an optimal control problem with fixed end point and fixed final time. Its objective is to compute an input profile that brings system states $\boldsymbol{x}_0$ reach a desired state $\boldsymbol{x}_1$ in t_1 time units:

The cost functional J consists of the controller objective, which could be tracking of the output $\boldsymbol{x}$ related to a desired set point, minimal usage of control actions $\boldsymbol{u}$, among other possibilities. This cost functional consists of stage functional $l(\boldsymbol{x}(t),\boldsymbol{u}(t))$ and a possible additional term called terminal functional $\boldsymbol{f}_{terminal}(t_1,\boldsymbol{x}(t_1),\boldsymbol{u}(t_1))$. This optimization problem minimizes the cost functional J while the dynamics of the system to be controlled are represented by a nonlinear model, as per constraint (4.1b). Constraints on the states (4.1e) and inputs (4.1f) can also apply.

The solution to Problem $\mathscr{P}_1$ consists of the optimal control policy $\boldsymbol{u}^*(t)$ which results in the optimal trajectory $\boldsymbol{x}^*(t)$.

Problem $\mathscr{P}_1$:

$$\min_{\boldsymbol{u}(t)} \quad J(\boldsymbol{x},\boldsymbol{u}) \tag{4.1}$$

where

$$J(\boldsymbol{x},\boldsymbol{u}) = \int_0^{t_1} l(\boldsymbol{x}(t),\boldsymbol{u}(t))dt + \boldsymbol{f}_{terminal}(t_1,\boldsymbol{x}(t_1),\boldsymbol{u}(t_1)), \tag{4.1a}$$

subject to

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{4.1b}$$

$$\boldsymbol{u}(0) = \boldsymbol{u}_0, \tag{4.1c}$$

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t),\boldsymbol{u}(t)), \tag{4.1d}$$

$$\boldsymbol{x}(t) \in \mathscr{X}, \tag{4.1e}$$

$$\boldsymbol{u}(t) \in \mathscr{U}, \tag{4.1f}$$

$$\boldsymbol{x}(t_1) = \boldsymbol{x}_1. \tag{4.1g}$$

4.2 Finite horizon optimal control problems

The controller is represented by a finite horizon optimal control problem (FHOCP) that is solved at every sampling interval. A not necessarily optimal sequence of control actions is computed. The first control action calculated is then implemented. The optimization problem is solved again at the next sampling time, while inputs and states (or outputs) are measured, which is why this control strategy may be referred to, in the literature, as moving horizon controller. The optimization problem is as follows:

Unlike in Problem $\mathscr{P}_1$, the optimal control problem that represents this controller is in discrete-time. This change happens due to most of the optimal control theory being presented in continuous time while the model predictive control literature handles sampled systems. The representation of states and inputs with a counter j as its argument, instead of a time variable t, is commonly used in the literature. All of the formulations in this work are in discrete-time.

In this optimization, the prediction horizon N_p and the control horizon N_c, seen in the objective function (4.2a), represent respectively how many sampling intervals there will be penalization of state error and control moves.

The stage cost, earlier represented by $l(\boldsymbol{x},\boldsymbol{u})$, consists now of penalization of state error with respect to a set point, $\boldsymbol{x}(k+j|k) - \boldsymbol{x}_{sp}$, as well as penalization of control moves, $\Delta\boldsymbol{u}(k+j|k)$, in order penalize more drastic input moves, increasing longevity of plant actuators. This penalization relative to a set point is not often found in the control theory literature, where it is assumed that the reference is the origin. However, there are no losses when one changes the reference from the origin to another point in the state space. Terminal penalty terms can also apply. This kind of controller is often referred to as a tracking controller, as it penalizes distance between state and a set-point $\boldsymbol{x}_{sp}$.

Problem $\mathscr{P}_2$:

$$\min_{\boldsymbol{u}_k} \quad J(\boldsymbol{x},\boldsymbol{u}), \tag{4.2}$$

where

$$J(\boldsymbol{x},\boldsymbol{u}) = \sum_{j=0}^{N_p-1} \left\|\boldsymbol{x}(k+j|k) - \boldsymbol{x}_{sp}\right\|^2_{\boldsymbol{Q}_x} + \sum_{j=0}^{N_c-1} \|\boldsymbol{\Delta u}(k+j|k)\|^2_{\boldsymbol{R}}, \tag{4.2a}$$

subject to

$$\boldsymbol{x}(k|k) = \boldsymbol{x}_0, \tag{4.2b}$$

$$\boldsymbol{u}(k-1|k) = \boldsymbol{u}_0, \tag{4.2c}$$

$$\boldsymbol{x}(k+j+1|k) = \boldsymbol{f}(\boldsymbol{x}(k+j|k),\boldsymbol{u}(k+j|k)), \tag{4.2d}$$

$$\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{u}(k+j|k) - \boldsymbol{u}(k+j-1|k), \tag{4.2e}$$

$$\boldsymbol{x}(k+j|k) \in \mathscr{X}, \tag{4.2f}$$

$$\boldsymbol{u}(k+j|k) \in \mathscr{U}, \tag{4.2g}$$

$$\boldsymbol{\Delta u}(k+j|k) \in \mathscr{U}_\Delta, \tag{4.2h}$$

$$\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{0}, \quad j \geq N_c, \tag{4.2i}$$

$$j \in [0, N_p-1] \in \mathbb{N}. \tag{4.2j}$$

The current values of state and input are represented respectively by constraints (4.2b) and (4.2c). Computation of input moves is handled by constraint (4.2e).

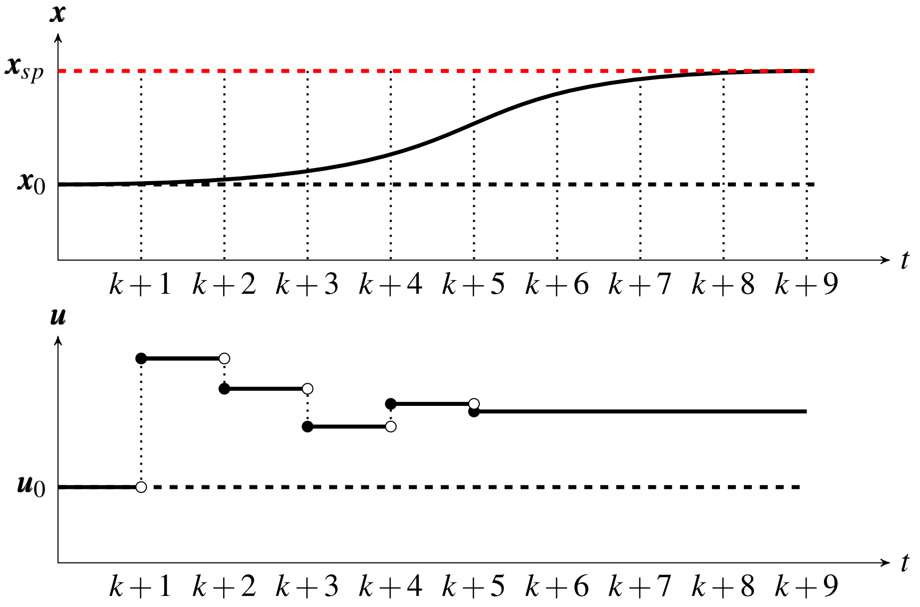
Admissible state, input, and input moves are represented by hard constraints (4.2f), (4.2g), and (4.2h). Constraint (4.2i) limits the number of control moves to compose the optimal input profile $\boldsymbol{u}_k$ to the number of control moves one can fit inside the control horizon. The final constraint (4.2j) indicates that the discrete-time counter j can only admit values inside the prediction horizon of length N_p.

The quadratic form of the stage cost observed in most of the stability theory literature is justified in Hahn (1967). In Chapter IV, prior to the direct Lyapunov method studies, he shows that the weighted Euclidean norm, named form in the mentioned work, is a convenient function to work with in stability studies since one can affirm the norm's positive definiteness or its radial unboundedness (relevant to global stability studies).

For illustrative purposes, Figure 19 demonstrates an open-loop trajectory predicted by the NMPC, along with its input profile.

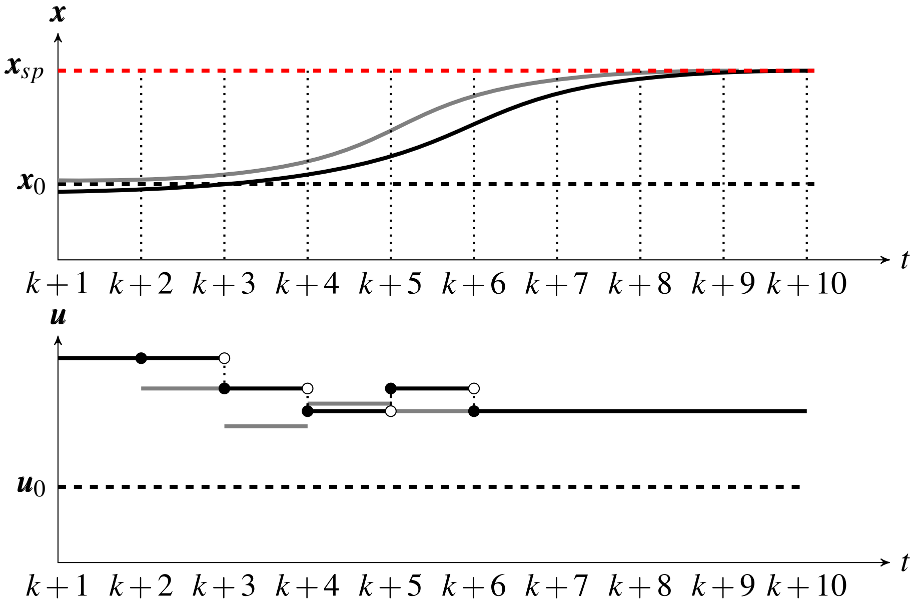
The open-loop system is at an equilibrium point $(\boldsymbol{x}_0, \boldsymbol{u}_0)$. The closed-loop trajectory consists of five control actions, the decision variable $\boldsymbol{u}_k = [\boldsymbol{u}(k+1|k) \; \ldots \; \boldsymbol{u}(k+5|k)]^T$ of Problem $\mathscr{P}_2$, and nine state predictions (therefore, $N_c = 5$ and $N_p = 9$), the state trajectory $[\boldsymbol{x}(k+1|k) \; \ldots \; \boldsymbol{x}(k+9|k)]^T$. This open-loop trajectory tracks a reference value $\boldsymbol{x}_{sp}$ as the optimization problem penalizes state error with respect to said reference.

Figure 19 – State and input profiles predicted and computed by NMPC in an iteration



Source: Own author.

Figure 20 – State prediction over time and computed inputs over time, computed via NMPC at previous (—) and current (—) iteration.



Source: Own author.

However, suppose there was a disturbance which brought state to $\boldsymbol{x}(k+1)$ instead of its predicted value at the previous sampling interval. Figure 20 illustrates this situation. Notice that there is a difference between the predicted states at the previous and the latter sampling intervals due to the disturbance, and that the computed control actions are different because the controller accounts for said disturbance. A different control profile is computed as the controller attempts to compensate for the state disturbance.

NMPC is a convenient control strategy for its capability of adapting to the current measurements of the plant. However, without any guarantee that it will stabilize the plant, the controller is not safe nor advantageous for its practical implementation.

The controller just exposed, in Problem $\mathscr{P}_2$, has no proven stability as is. Studies in this regard revolve around finding conditions where this formulation is stabilizing, either via stabilizing constraints (by adding a constraint to the optimization problem, or manipulating the weighting of the objective function) or control system characteristics (meaning no modifications are necessary in order to achieve closed-loop stability, instead stability is achieved through controller tuning alone).

4.2.1 Stability granting methods

Stabilizing formulations found in the literature will be presented, followed by examples - when possible.

For linear systems, infinite horizon model predictive controllers are nominally stable (MUSKE; RAWLINGS, 1993). An optimization problem that may represent such controller for a nonlinear system is:

Problem $\mathscr{P}_3$:

$$\min_{\boldsymbol{u}_k} \quad J(\boldsymbol{x},\boldsymbol{u}) \tag{4.3}$$

where

$$J(\boldsymbol{x},\boldsymbol{u}) = \sum_{j=0}^{\infty} \|\boldsymbol{x}(k+j|k) - \boldsymbol{x}_{sp}\|^2_{\boldsymbol{Q}_x} + \sum_{j=0}^{\infty} \|\boldsymbol{\Delta u}(k+j|k)\|^2_{\boldsymbol{R}}, \tag{4.3a}$$

subject to

$$\boldsymbol{x}(k|k) = \boldsymbol{x}_0, \tag{4.3b}$$
$$\boldsymbol{u}(k-1|k) = \boldsymbol{u}_0, \tag{4.3c}$$
$$\boldsymbol{x}(k+j+1|k) = \boldsymbol{f}(\boldsymbol{x}(k+j|k),\boldsymbol{u}(k+j|k)), \tag{4.3d}$$
$$\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{u}(k+j|k) - \boldsymbol{u}(k+j-1|k), \tag{4.3e}$$
$$\boldsymbol{x}(k+j|k) \in \mathscr{X}, \tag{4.3f}$$
$$\boldsymbol{u}(k+j|k) \in \mathscr{U}, \tag{4.3g}$$
$$\boldsymbol{\Delta u}(k+j|k) \in \mathscr{U}_\Delta, \tag{4.3h}$$
$$j \in \mathbb{N}. \tag{4.3i}$$

The infinite dimension of the decision variable, the input trajectory $\boldsymbol{u}_k$, makes the problem intractable. Also, the infinite summations in the cost function are unbounded. The optimization problem's intractability makes it impossible to implement such controller. Attempts to handle this intractability of infinite horizon controllers first came from the use of equality constraints (KEERTHI; GILBERT, 1988).

4.2.1.1 Terminal state equality constraints

Restrictions can be added to the optimization problem in order to limit the summations, as well turning the decision variable into a finite sequence of control actions. Two restrictions are necessary, one for each infinite summation, resulting in Problem $\mathscr{P}_4$.

Problem $\mathscr{P}_4$:

$$\min_{\boldsymbol{u}_k} \quad J(\boldsymbol{x},\boldsymbol{u}), \tag{4.4}$$

where

$$J(\boldsymbol{x},\boldsymbol{u}) = \sum_{j=0}^{\infty} \left\|\boldsymbol{x}(k+j|k) - \boldsymbol{x}_{sp}\right\|_{\boldsymbol{Q}_x}^2 + \sum_{j=0}^{\infty} \|\boldsymbol{\Delta u}(k+j|k)\|_{\boldsymbol{R}}^2, \tag{4.4a}$$

subject to

$$\begin{aligned}
&\boldsymbol{x}(k|k) = \boldsymbol{x}_0, && (4.4b)\\
&\boldsymbol{u}(k-1|k) = \boldsymbol{u}_0, && (4.4c)\\
&\boldsymbol{x}(k+j+1|k) = \boldsymbol{f}(\boldsymbol{x}(k+j|k),\boldsymbol{u}(k+j|k)), && (4.4d)\\
&\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{u}(k+j|k) - \boldsymbol{u}(k+j-1|k), && (4.4e)\\
&\boldsymbol{x}(k+j|k) \in \mathscr{X}, && (4.4f)\\
&\boldsymbol{u}(k+j|k) \in \mathscr{U}, && (4.4g)\\
&\boldsymbol{\Delta u}(k+j|k) \in \mathscr{U}_\Delta, && (4.4h)\\
&\boldsymbol{x}(k+N_p-1|k) - \boldsymbol{x}_{sp} = \boldsymbol{0}, && (4.4i)\\
&\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{0},\ j \geq N_c, && (4.4j)\\
&\boldsymbol{f}(\boldsymbol{x}(k+N_p-1|k),\boldsymbol{u}(k+N_c-1|k)) = \boldsymbol{x}(k+N_p-1|k), && (4.4k)\\
&j \in [0, N_p-1] \in \mathbb{N}. && (4.4l)
\end{aligned}$$

With (4.4i) and (4.4j), it can be shown that the infinite summations become limited. We split the infinite summation in two, referring to the summation inside the controller's horizon and outside said horizon.

$$\begin{aligned}
\sum_{j=0}^{\infty} \left\|\boldsymbol{x}(k+j|k) - \boldsymbol{x}_{sp}\right\|_{\boldsymbol{Q}_x}^2 + \sum_{j=0}^{\infty} \|\boldsymbol{\Delta u}(k+j|k)\|_{\boldsymbol{R}}^2 =& \\
\sum_{j=0}^{N_p-1} \left\|\boldsymbol{x}(k+j|k) - \boldsymbol{x}_{sp}\right\|_{\boldsymbol{Q}_x}^2 + \sum_{j=0}^{N_c-1} \|\boldsymbol{\Delta u}(k+j|k)\|_{\boldsymbol{R}}^2& \\
+ \sum_{j=0}^{\infty} \left\|\boldsymbol{x}(k+N_p+j|k) - \boldsymbol{x}_{sp}\right\|_{\boldsymbol{Q}_x}^2 + \sum_{j=0}^{\infty} \|\boldsymbol{\Delta u}(k+N_c+j|k)\|_{\boldsymbol{R}}^2&
\end{aligned}$$

When the controller satisfies constraints (4.4i) and (4.4j), we have that the infinite summations on the right hand side of the equality above are equal to zero, making the infinite summation bounded. This method of eliminating the intractability of infinite summations has been done first by Keerthi & Gilbert (1988). Constraint (4.4k) guarantees that only trajectories that reach steady state at the end of the prediction horizon are considered.

The formulation was implemented for a quadruple tank system. Tuning parameters consist of $\boldsymbol{Q}_x = 1 \times I_{n_x}$, $\boldsymbol{Q}_u = 1 \times I_{n_u}$, $\boldsymbol{R} = 1 \times I_{n_u}$, $N_P = N_C = 27$, with sampling time of 2 seconds. The closed-loop states can be seen in Figure 21. The inputs computed can be seen in Figure 22. The controller cost is seen in Figure 23. On the optimization side of the simulation, the output flag and the computer time spent per sampling interval can be seen in Figure 24.

The formulation represented by Problem $\mathscr{P}_4$ tracks different set points properly. More drastic input moves can be seen in the latter third part of the simulation. However, the input move constraints are not yet violated. The controller cost function shows non-increasing behavior. The optimization was completed successfully throughout the entire simulation. A slight increase in computational effort can be seen at the start of the third part of the simulation, where the controller exerts the largest control effort in order to satisfy the terminal equality constraints.

Figure 21 – Tracking case of $\mathscr{P}_4$: states (—) and reference (- - -)

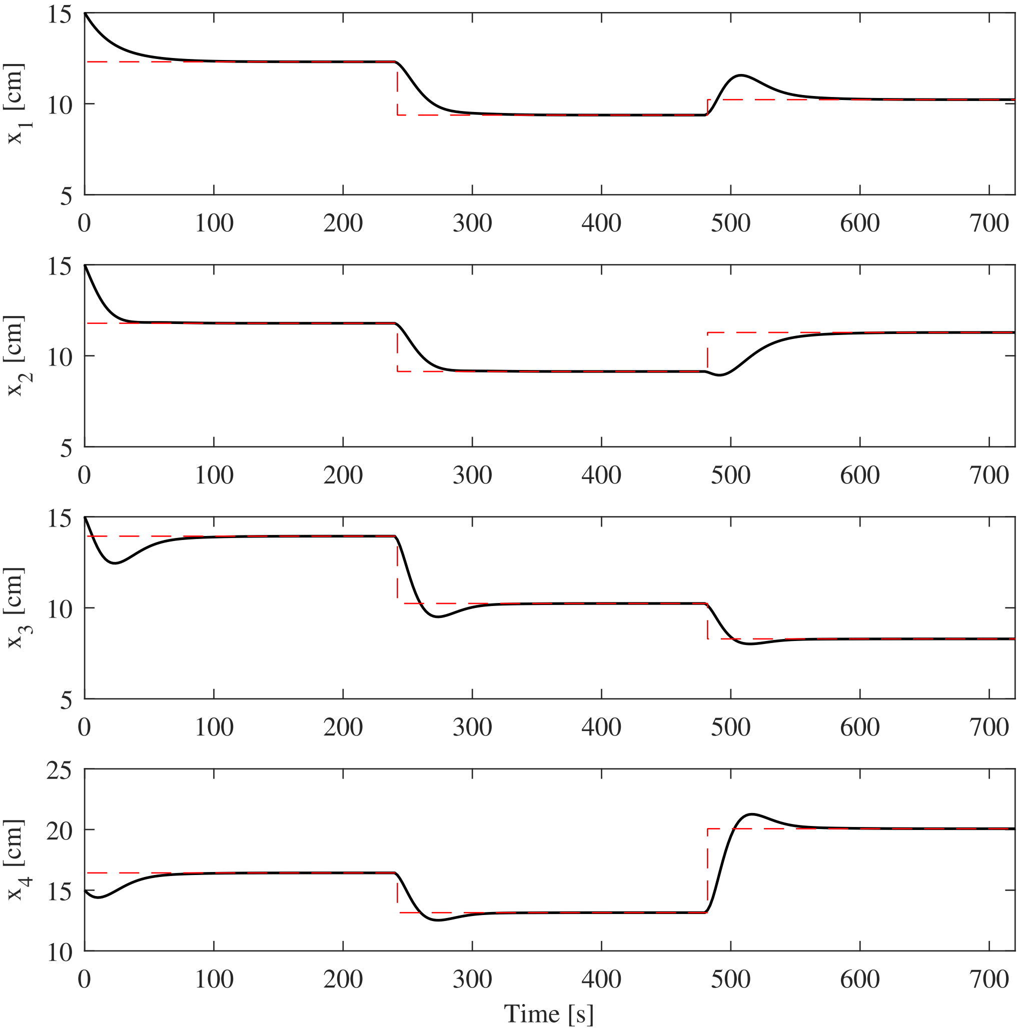


Figure 22 – Tracking case of $\mathscr{P}_4$: inputs

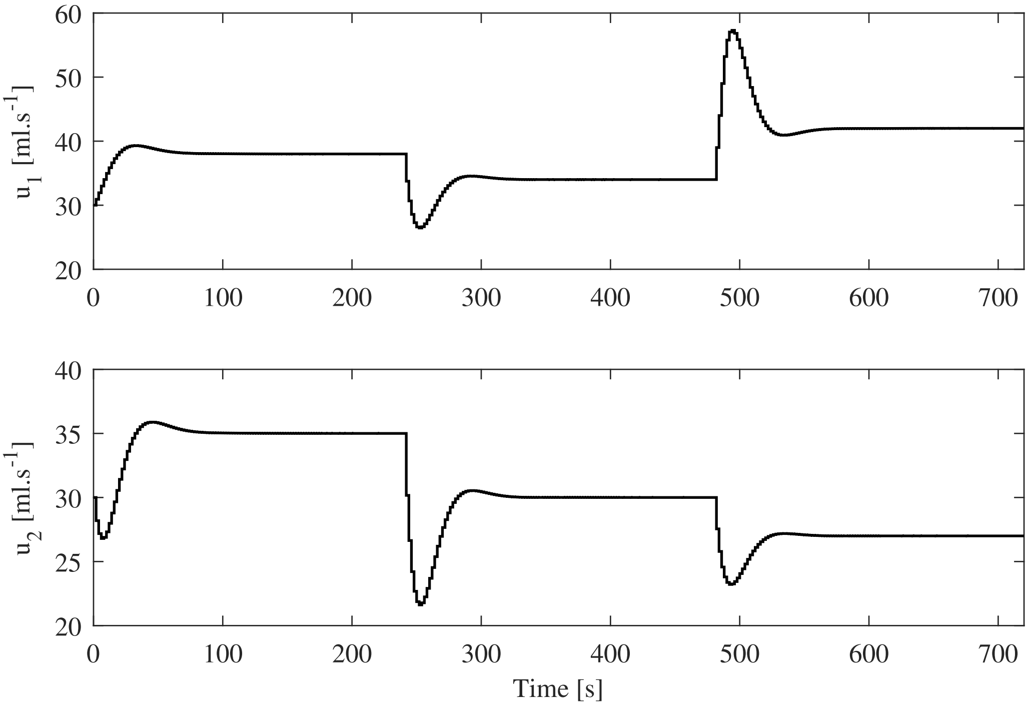


Figure 23 – Tracking case of $\mathscr{P}_4$: cost

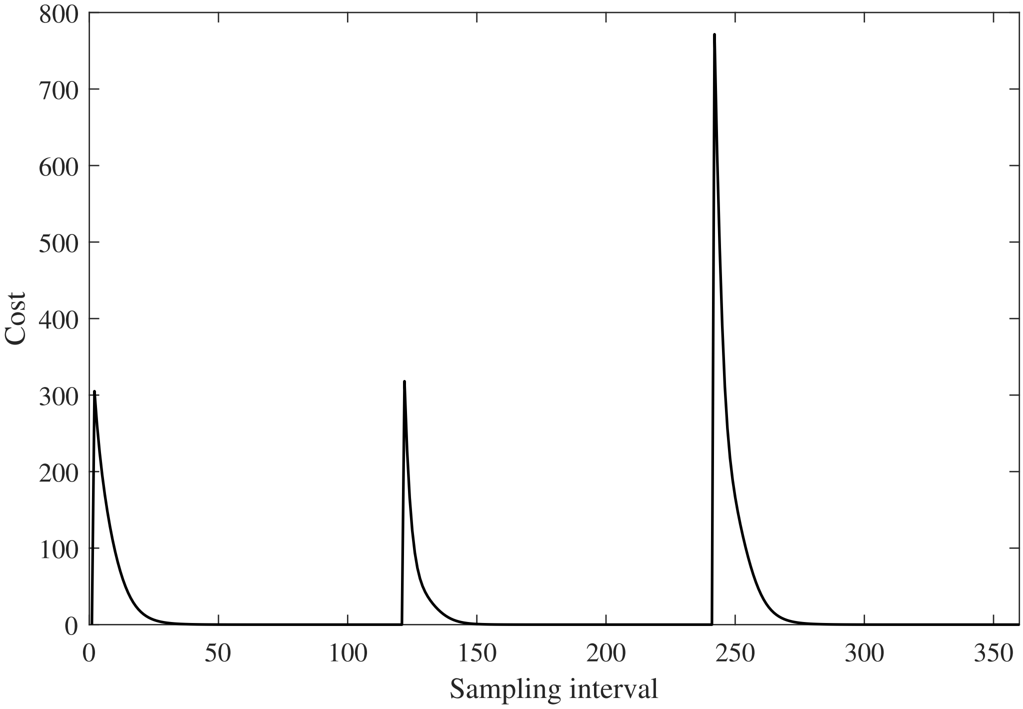
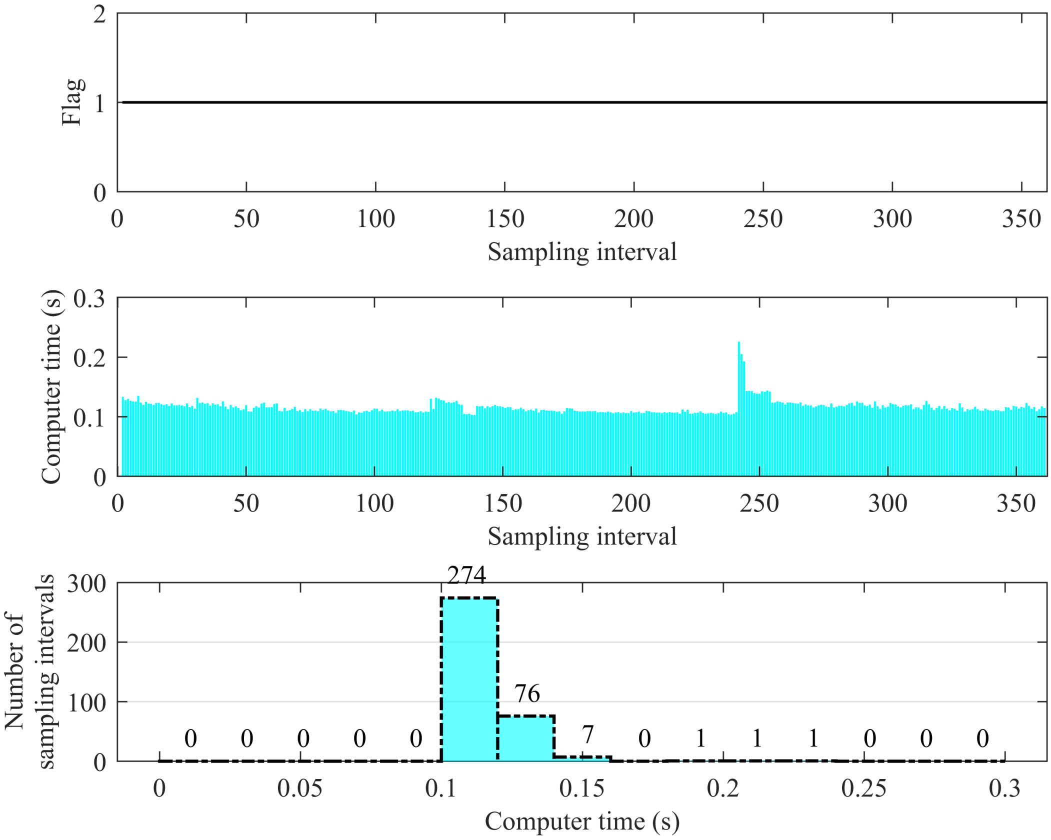


Figure 24 – Tracking case of $\mathscr{P}_4$: flag and computer times



The terminal equality constraint will be implemented in this work along with zone control with optimizing targets. (GONZÁLEZ; ODLOAK, 2009) Zone control has as objective bringing states to a range of operation instead of a set point. With this artificial reference free to be chosen by the controller, one can then impose a penalization of inputs with respect to an optimal input target. A finite horizon optimal control problem (FHOCP) that represents this formulation is Problem $\mathscr{P}_5$.

Changes from the original FHOCP $\mathscr{P}_2$, that represents the traditional NMPC, are introduction of the set point and its corresponding input as decision variables, that must satisfy an additional constraint (4.5i), representing control zone. The constraints (4.5j) and (4.5k) are added in order to have an equilibrium point as a reference value with feasible corresponding input.

Penalization of this input with respect to a desired input is an economic term that can be added to the cost function, as done in (4.5a). This can be used to either minimize or maximize input values, or to keep inputs closest to an optimal target.

Integration of terminal equality constraints and zone control has their recursive feasibility and asymptotic stability demonstrated, respectively, in Lemma 1 and Theorem 1 of Sencio et al.

(2020). The optimization problem that integrates a terminal equality constraint formulation and zone control with optimizing targets is given by Problem $\mathscr{P}_6$.

Problem $\mathscr{P}_5$:

$$\min_{\boldsymbol{u}_k,\boldsymbol{x}_{sp},\boldsymbol{u}_{sp}} \quad J(\boldsymbol{x},\boldsymbol{u}), \tag{4.5}$$

where

$$\begin{aligned} J(\boldsymbol{x},\boldsymbol{u}) = \sum_{j=0}^{N_p-1} \left\|\boldsymbol{x}(k+j|k) - \boldsymbol{x}_{sp}\right\|_{\boldsymbol{Q}_x}^2 + \sum_{j=0}^{N_c-1} \|\boldsymbol{\Delta u}(k+j|k)\|_{\boldsymbol{R}}^2 \\ + \left\|\boldsymbol{u}_{sp} - \boldsymbol{u}_{des}\right\|_{\boldsymbol{Q}_u}^2, \end{aligned} \tag{4.5a}$$

subject to

$$\boldsymbol{x}(k|k) = \boldsymbol{x}_0, \tag{4.5b}$$
$$\boldsymbol{u}(k-1|k) = \boldsymbol{u}_0, \tag{4.5c}$$
$$\boldsymbol{x}(k+j+1|k) = \boldsymbol{f}(\boldsymbol{x}(k+j|k),\boldsymbol{u}(k+j|k)), \tag{4.5d}$$
$$\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{u}(k+j|k) - \boldsymbol{u}(k+j-1|k), \tag{4.5e}$$
$$\boldsymbol{x}(k+j|k) \in \mathscr{X}, \tag{4.5f}$$
$$\boldsymbol{u}(k+j|k) \in \mathscr{U}, \tag{4.5g}$$
$$\boldsymbol{\Delta u}(k+j|k) \in \mathscr{U}_\Delta, \tag{4.5h}$$
$$\boldsymbol{x}_{sp} \in \mathscr{X}_Z, \tag{4.5i}$$
$$\boldsymbol{x}_{sp} \in \mathscr{X}_\omega, \tag{4.5j}$$
$$\boldsymbol{u}_{sp} \in \mathscr{U}, \tag{4.5k}$$
$$\boldsymbol{x}_{sp} = \boldsymbol{f}(\boldsymbol{x}_{sp},\boldsymbol{u}_{sp}), \tag{4.5l}$$
$$j \in [0, N_p - 1] \in \mathbb{N}. \tag{4.5m}$$

Here, we prove the recursive feasibility of the formulation in Problem $\mathscr{P}_6$:

Lemma 2 *For an undisturbed nominal system, if Problem $\mathscr{P}_6$ is feasible at time step k, it will remain feasible for any subsequent time step.*

With the assumption that Problem $\mathscr{P}_6$ is feasible at k, there is an optimal control sequence

$$\boldsymbol{u}_k^* = [\boldsymbol{u}^*(k|k)^T \ \dots \ \boldsymbol{u}^*(k+N_c-2|k)^T \ \boldsymbol{u}^*(k+N_c-1|k)^T]^T.$$

With it, one can build a candidate solution for the next sampling time $k+1$ which is

$$\tilde{\boldsymbol{u}}_{k+1} = [\boldsymbol{u}^*(k+1|k)^T \ \dots \ \boldsymbol{u}^*(k+N_c-1|k)^T \ \boldsymbol{u}^*(k+N_c-1|k)^T]^T.$$

Input constraints (4.6g) and (4.6h) are not violated by this candidate solution. Also, as the candidate solution shifts the optimal input trajectory $\boldsymbol{u}_k^*$ with repeated control action - no

Problem $\mathscr{P}_6$:

$$\min_{\boldsymbol{u}_k,\boldsymbol{x}_{sp},\boldsymbol{u}_{sp}} \quad J(\boldsymbol{x},\boldsymbol{u}), \tag{4.6}$$

where

$$\begin{aligned} J(\boldsymbol{x},\boldsymbol{u}) = & \sum_{j=0}^{N_p-1} \left\|\boldsymbol{x}(k+j|k)-\boldsymbol{x}_{sp}\right\|^2_{\boldsymbol{Q}_x} + \sum_{j=0}^{N_c-1} \|\boldsymbol{\Delta u}(k+j|k)\|^2_{\boldsymbol{R}} \\ & + \left\|\boldsymbol{u}_{sp}-\boldsymbol{u}_{des}\right\|^2_{\boldsymbol{Q}_u}, \end{aligned} \tag{4.6a}$$

subject to

$$\boldsymbol{x}(k|k) = \boldsymbol{x}_0, \tag{4.6b}$$
$$\boldsymbol{u}(k-1|k) = \boldsymbol{u}_0, \tag{4.6c}$$
$$\boldsymbol{x}(k+j+1|k) = \boldsymbol{f}(\boldsymbol{x}(k+j|k),\boldsymbol{u}(k+j|k)), \tag{4.6d}$$
$$\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{u}(k+j|k)-\boldsymbol{u}(k+j-1|k), \tag{4.6e}$$
$$\boldsymbol{x}(k+j|k) \in \mathscr{X}, \tag{4.6f}$$
$$\boldsymbol{u}(k+j|k) \in \mathscr{U}, \tag{4.6g}$$
$$\boldsymbol{\Delta u}(k+j|k) \in \mathscr{U}_\Delta, \tag{4.6h}$$
$$\boldsymbol{x}_{sp} \in \mathscr{X}_Z, \tag{4.6i}$$
$$\boldsymbol{x}_{sp} \in \mathscr{X}_\omega, \tag{4.6j}$$
$$\boldsymbol{u}_{sp} \in \mathscr{U}, \tag{4.6k}$$
$$\boldsymbol{x}(k+N_p-1|k)-\boldsymbol{x}_{sp} = \boldsymbol{0}, \tag{4.6l}$$
$$\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{0},\ j \geq N_c \tag{4.6m}$$
$$\boldsymbol{f}(\boldsymbol{x}_{sp},\boldsymbol{u}(k+N_c-1|k)) = \boldsymbol{x}_{sp} \tag{4.6n}$$
$$j \in [0,N_p] \in \mathbb{N}. \tag{4.6o}$$

control move is done - taken at its final element, the candidate solution for the reference value is the previously computed optimal reference value, $\tilde{\boldsymbol{x}}_{sp} = \boldsymbol{x}^*_{sp}$. Also due to no control action taken at the end of the candidate input trajectory $\tilde{\boldsymbol{u}}_k$, the terminal equality constraint (4.6l) and the steady-state constraint (4.6n) are also satisfied. With the terminal equality constraint (4.6l) satisfied at k, this means that the final state $\boldsymbol{x}(k+N_p-1|k)$ is a steady state. As no control action is taken after $k+N_c-1$, the candidate solution yields another steady state at the end of its prediction, $\boldsymbol{x}(k+N_c-1|k+1) = \boldsymbol{x}(k+N_c-1|k)$, satisfying (4.6l) at $k+1$. ■

With recursive feasibility proven, we guarantee stability as follows:

Theorem 2 *Consider the nominal system in the absence of disturbances and let the intersection between the set of equilibrium points and the control zone $\mathscr{X}_\omega \cap \mathscr{X}_Z$ be non empty. Also, suppose that Problem $\mathscr{P}_6$ has a feasible solution at time step k and that $\boldsymbol{u}_{des}$ is an admissible input target. The controller drives the system into the control zone in which the pair $(\boldsymbol{x}_{sp},\boldsymbol{u}_{des})$, $\boldsymbol{x}_{sp} \in \mathscr{X}_\omega$, $\boldsymbol{u}_{des} \in \mathscr{U}$ is an asymptotically stable solution to Problem $\mathscr{P}_6$.*

The stability proof once again makes use of an optimal solution $\boldsymbol{u}_k^*$ and $\boldsymbol{x}_{sp}^*$ and an inherited solution $\tilde{\boldsymbol{u}}_{k+1}$ and $\tilde{\boldsymbol{x}}_{sp}$. The optimal solution produces an optimal cost J^* at sampling time k. At sampling time $k+1$, a sub-optimal solution which consists of the inherited solution $\tilde{\boldsymbol{u}}_{k+1}$ produces a sub-optimal cost $\tilde{J}$ at sampling time $k+1$. If one were to compare the costs $J^*(k)$ and $\tilde{J}(k+1)$, the difference between these would be:

$$J^*(k) - \tilde{J}(k+1) = \left\|\boldsymbol{x}(k|k) - \boldsymbol{x}_{sp}\right\|_{\boldsymbol{Q}_x}^2 + \|\boldsymbol{\Delta u}^*(k|k)\|_{\boldsymbol{R}}^2$$

With positive-definite matrices $\boldsymbol{Q}_x$ and $\boldsymbol{R}$, we have that this difference is non-negative, which implies that $\tilde{J}(k+1) \leq J^*(k)$. Since the inherited solution may be sub-optimal, this also implies that $J^*(k+1) \leq J^*(k)$. Consequently, future controller cost values will always be non-increasing. Built by norms, the future controller costs are also bounded below by zero. When the input target $\boldsymbol{u}_{des}$ is reachable, the cost converges to zero. With cost convergence to zero, there is convergence of state and input, respectively $\boldsymbol{x}(k) \rightarrow \boldsymbol{x}_{sp}$ and $\boldsymbol{u}(k) \rightarrow \boldsymbol{u}_{sp} = \boldsymbol{u}_{des}$ as $k \rightarrow \infty$. ■

Closed-loop simulations of a quadruple tank system controlled by Problem $\mathscr{P}_6$ were performed. Tuning parameters consist of $\boldsymbol{Q}_x = 1 \times I_{n_x}$, $\boldsymbol{Q}_u = 1 \times I_{n_u}$, $\boldsymbol{R} = 1 \times I_{n_u}$, $N_P = N_C = 27$, with sampling time of 2 seconds. The resulting states can be seen in Figure 25, whereas the computed control policies are seen in Figure 26, Figure 27 shows the controller cost and finally Figure 28 shows the optimization flags and computer time spent per iteration for this formulation.

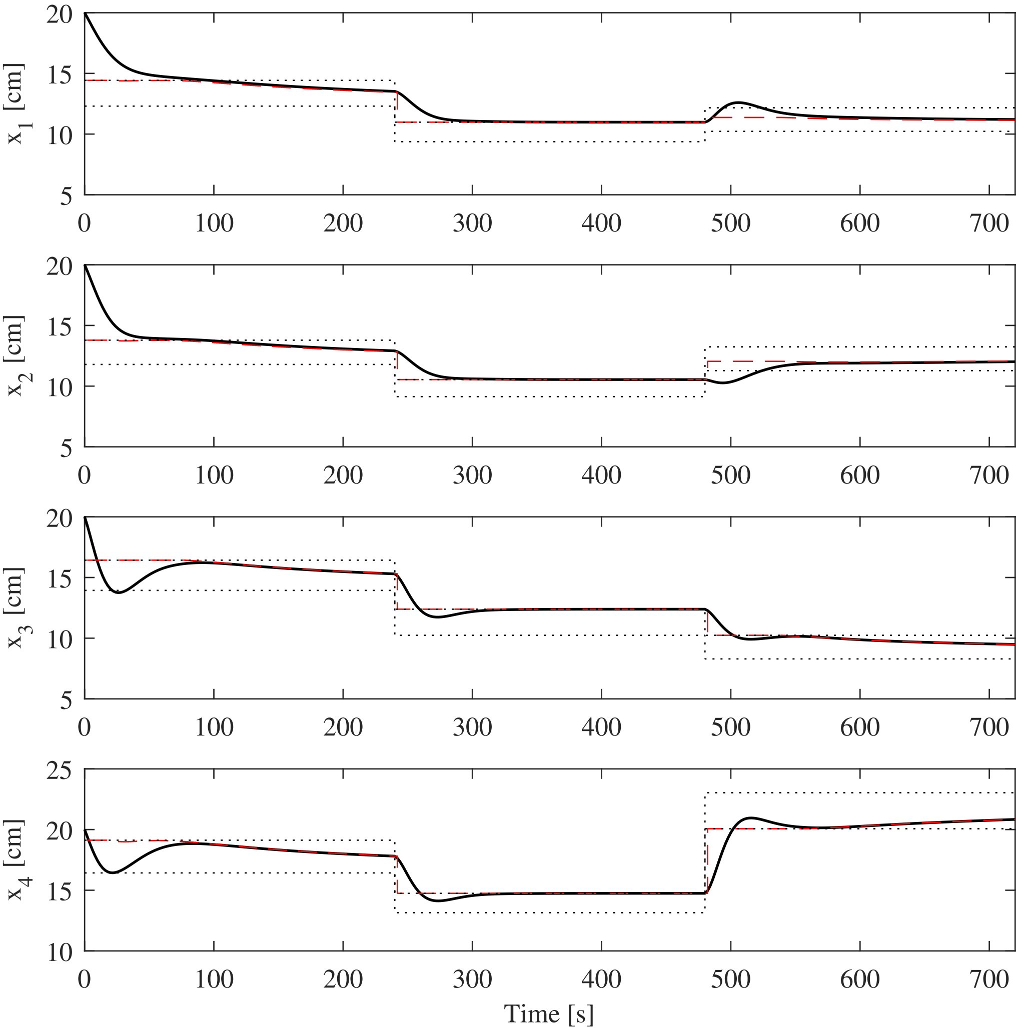
With the set point as a decision variable, it can be seen that the controller moves the system states inside the control zone. As the states enter the control zone, the next priority is to attain the input target. This is entirely due to the tuning parameters related to state error and input target error.

When the controller brings the states to the control zone, its next objective is to minimize the distance between input and its economic target. When the input target is not related to an equilibrium point inside the control zone, the controller then brings the state to the border of the control zone while minimizing the distance between input and its economic target. As per the objective function (4.6a), this results in non-zero cost while the closed-loop is stable.

As mentioned, the second part of the simulation has a non-zero cost due to the offset between input and its economic target. Overall, the controller cost shows non-increasing behavior as well.

The optimization was solved successfully at every sampling time. The computational effort in this case is higher than for the tracking case because the controller computes not only control actions but a set point and a corresponding input at every iteration. If one were to apply the rule of thumb of 10% of the sampling time for computer times, this controller would be considered inapplicable for the quadruple tank system due to multiple violations of said rule. Strategies that accelerate computer times will be discussed later.

Figure 25 – Zone control with input targets of $\mathscr{P}_6$: states (—), control zones (⋯) and computed setpoints (- - -)



Constraints (4.4i) and (4.4k) of Problem $\mathscr{P}_4$ - (4.6l) and (4.6n) for Problem $\mathscr{P}_6$ - are very restrictive and usually make the problem inapplicable due to longer horizon demands. Adaptive horizon methods can be employed, in order to compute the length of the horizon necessary in order to make these constraints feasible. Adaptive methods extend and diminish the control and prediction horizons depending on how far the controller predictions are from satisfying (4.4i) and (4.6l). One could preemptively compute the control and prediction horizons in order to always satisfy the equality constraints. This horizon tuning can be done through optimization, as was done by Giraldo, Melo & Secchi (2019). Another approach, based on multiple open-loop simulations can be used in order to determine a first guess to these horizon lengths. By the bang-bang principle - which states that the boundary of the attainable set is obtained through maximal control actions - along with a near steady-state condition, which is how optimization solvers handle equality constraints, one can evaluate if the system has reached the operating zone and is close to steady-state. The near steady-state condition would be of the form

Figure 26 – Zone control with input targets of $\mathscr{P}_6$: inputs (—) and input targets (- - -)

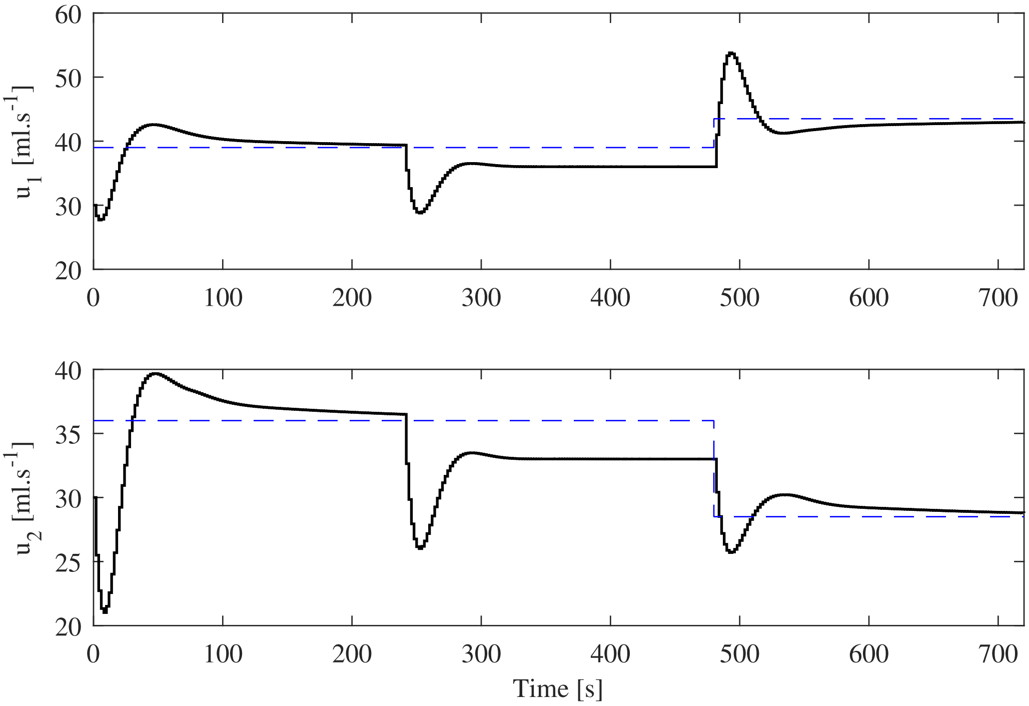
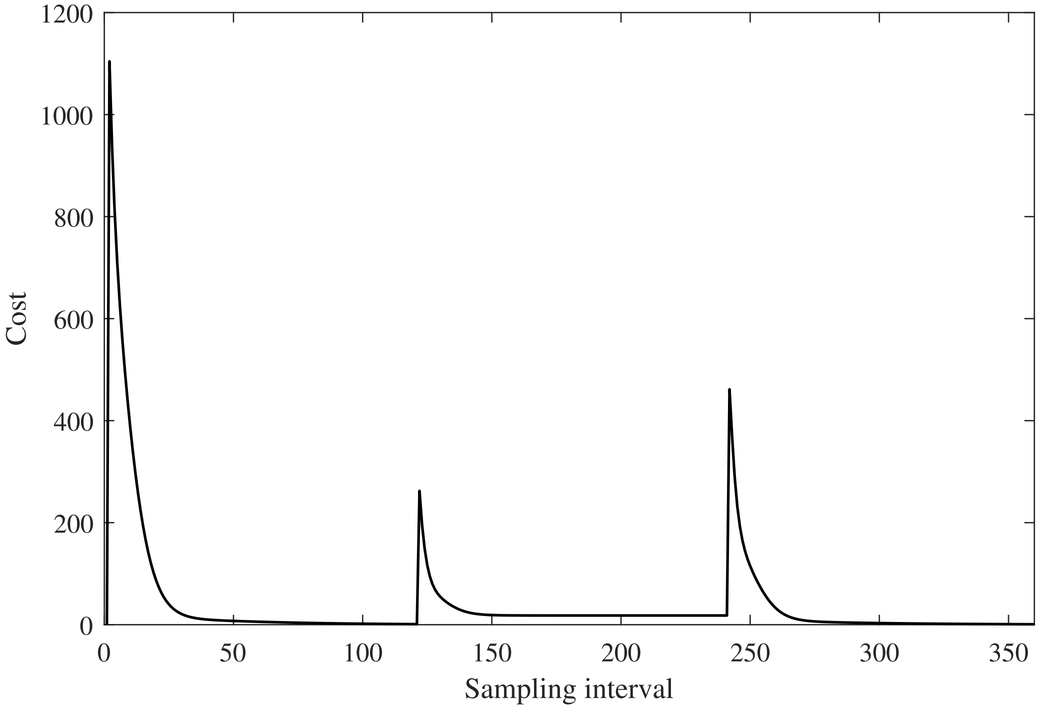


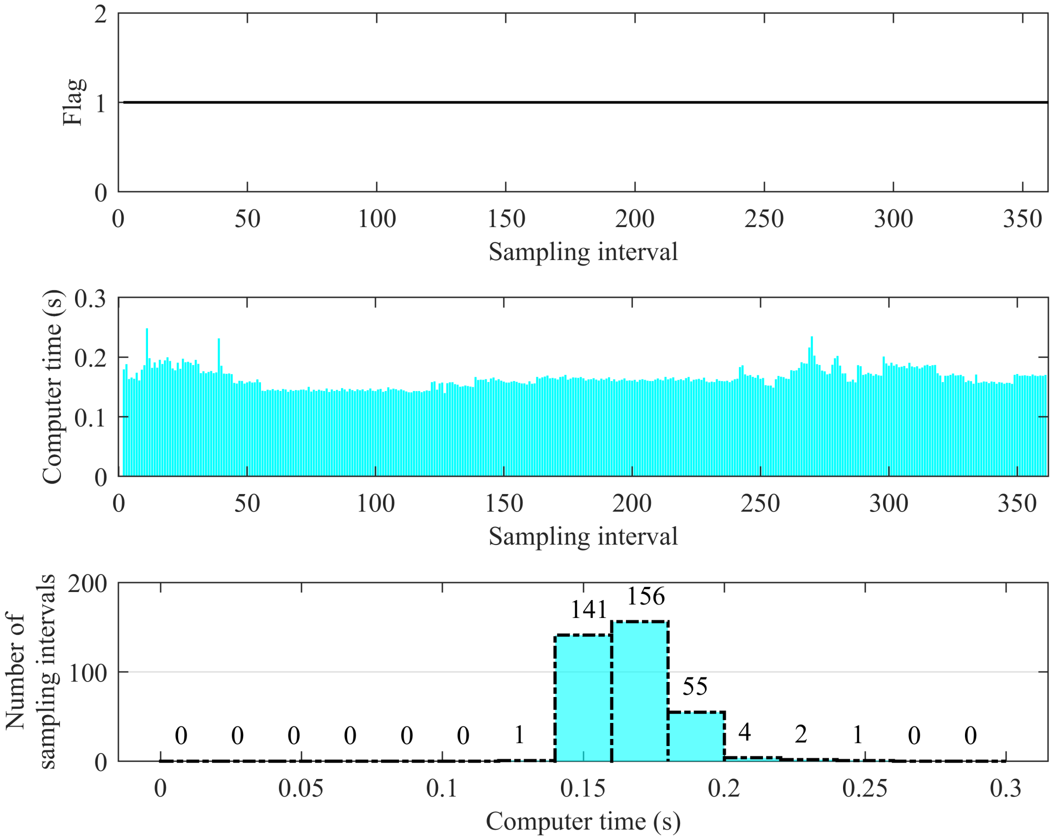
Figure 27 – Zone control with input targets of $\mathscr{P}_6$: cost



$$-\varepsilon\begin{bmatrix}1 & 1 & 1\end{bmatrix}^T \leq \dot{\boldsymbol{x}}(k+N_p-1|k) \leq \varepsilon\begin{bmatrix}1 & 1 & 1\end{bmatrix}^T \tag{4.7}$$

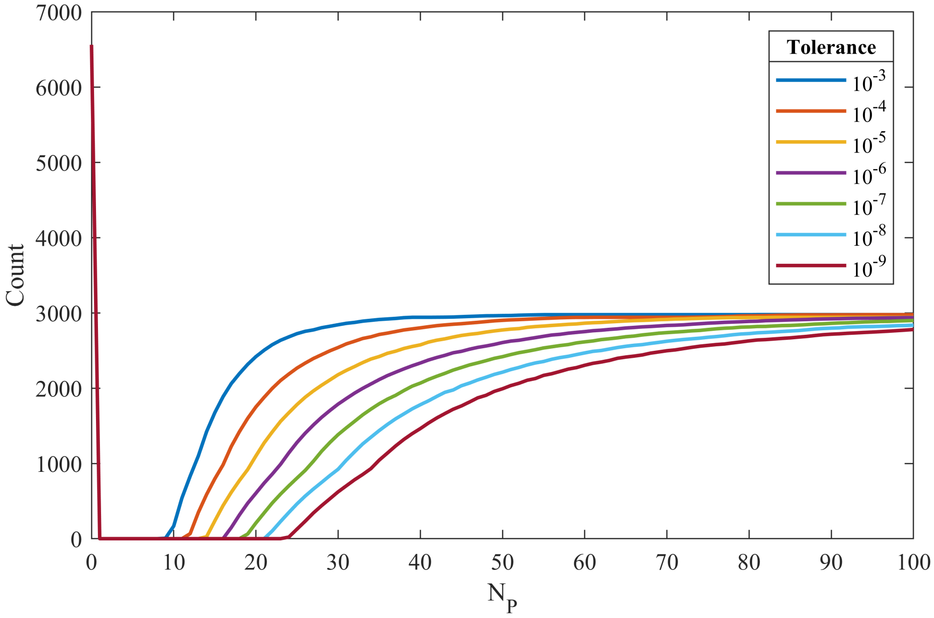
with ε as the steady-state tolerance. Considering once again the CSTR reactor (2.14), different values of tolerances were used in order to check how many different trajectories reach a specified zone and steady-state. The system in question was simulated in open-loop with control

Figure 28 – Zone control with input targets of $\mathscr{P}_6$: flag and computer times



actions at the border of the admissible input set and considering input moves at the border of the input move set as well. For the same initial state, over 6000 open-loop simulations were performed in order to detect which control actions would result in trajectories that reach the control zone while attending the steady-state tolerance. The results are available in Figure 29.

Figure 29 – Trajectory count that reach control zone by prediction horizon length for multiple steady-state tolerances



Source: Own author.

With Figure 29, one can appropriately choose the prediction horizon N_p according to the desired steady-state tolerance ε. Not every trajectory reaches the desired control zone, but that is expected since not every control move will reach the control zone. As one decreases the steady-state tolerance, less trajectories are available. With longer horizons, more trajectories are capable of satisfying the steady-state tolerance.

One can make use of unbound slack variables on said constraints in order to make these constraints always feasible. One must penalize these slacks as well in the cost functional, otherwise a static state trajectory may be taken rather than a trajectory that goes towards a set point or zone. This strategy, as is, has not yet been proven to have guaranteed stability or recursive feasibility. The work of Fagiano & Teel (2012) has circumvented the need of a terminal state equality constraint by means of a terminal state penalization that is tuned by the operator and is made to contract at every sampling interval. Later in this work a strategy that makes use of slack variables will be implemented and its guaranteed stability and recursive feasibility demonstrated.

More strategies have been developed in the literature, to approximate infinite horizon controllers. The most relevant ones are the cost-contracting constraint, state-contracting constraint and terminal inequality constraints.

4.2.1.2 Terminal state inequality constraints

The use of a linear quadratic regulator (LQR) was proposed by Michalska & Mayne (1993). A dual-mode controller that consists of a model predictive controller that drives the system to a region about the desired value where a stabilizing linear quadratic regulator operates.

Chen & Allgöwer (1998) introduce a quasi-infinite horizon NMPC. The objective functional consists of a finite horizon cost which penalizes the states through the finite horizon and a terminal cost which represents the controller cost after the control horizon. The terminal cost is computed via cost of a LQR which is never active, like the classic dual-mode MPC (MICHALSKA; MAYNE, 1993). The terminal inequality constraint represents the restriction of the terminal state to a region where such LQR is stabilizing. A discussion of the rationale behind the terminal region and corresponding control law follows, before advancing to the controller itself.

An alternative terminal region MPC application involving polytopic invariant sets (CANNON; DESHMUKH; KOUVARITAKIS, 2003), instead of ellipsoidal invariant sets, must be mentioned, although it was not implemented in this work.

As continuous-time formulations are difficult to implement online due to its fast sampling requirements, we proceed to a formulation for discrete-time nonlinear systems. The final QIH-NMPC to be shown here consists of determining the discrete LQR for the terminal region and working with the nonlinear system in discrete-time. Depending on sampling rates, continuous-time formulations are not viable since the computed inputs injected to the plant cannot approximate properly a continuous-time optimal input profile.

Rajhans et al. (2019) extended the work of Chen & Allgöwer (1998) to discrete-time systems. Its workflow is very similar: compute the regulator and terminal penalty matrix through the discrete algebraic Ricatti equation (DARE), identify the terminal region where input constraints are satisfied then finally identify the region inside this terminal region where the nonlinearity is bounded, hence having the nonlinear system stabililized by the linear state feedback control law computed by DARE. The terminal penalty matrix and terminal region are used as stabilizing constraints in the NMPC, which has its closed-loop stability given by recursive feasibility and guaranteed convergence proven in a similar fashion as the original QIH-NMPC.

Consider the discrete-time nonlinear system:

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{4.8}$$

and a different set of assumptions:

Assumption 1 $\boldsymbol{f} : \mathscr{X} \times \mathscr{U} \rightarrow \mathscr{X}$ *is continuously differentiable over the set of admissible states* $\mathscr{X}$.

Assumption 2 *$\mathscr{U}$ is compact and contains the reference value in its interior.*

Assumption 3 *The set of admissible states $\mathscr{X}$ is control positive invariant for $\boldsymbol{f}(\cdot,\cdot)$, is compact and contains the reference value in its interior.*

This is a new class of assumptions, prevalent in more recent predictive control literature. Control positive invariance means that any admissible state cannot evolve outside the set through any admissible inputs, i.e. $\boldsymbol{f}(\boldsymbol{x}(k),\boldsymbol{u}(k) \in \mathscr{X}$ for all $\boldsymbol{x}(k) \in \mathscr{X}$ and all $\boldsymbol{u}(k) \in \mathscr{U}$ and $k > 0$.

Assumption 4 *The reference value $(\boldsymbol{x}_{sp},\boldsymbol{u}_{sp})$ is an equilibrium point of the system.*

That is, $\boldsymbol{f}(\boldsymbol{x}_{sp},\boldsymbol{u}_{sp}) = \boldsymbol{x}_{sp}$.

Assumption 5 *States $\boldsymbol{x}(k)$, $k \geq 0$, are perfectly measured and the system has no disturbances.*

Representing the standard undisturbed nominal case.

Assumption 6 *The Jacobian linearization of the nonlinear system is controllable.*

Necessary assumption since the virtual regulator needs for asymptotic stability guarantee.

We proceed to show the optimization problem for the QIH-NMPC for the discrete-time system:

Recursive feasibility is stated by the following Lemma.

Lemma 3 (adapted from Lemma 13 of Rajhans et al. (2019)) *For the nominal undisturbed system with perfect measurement, when the QIH-NMPC formulation represented by Problem $\mathscr{P}_7$ is feasible for $k = 0$, it is feasible for any $k > 0$.*

The proof begins with the optimal solution obtained at time $k = 0$:

$$\boldsymbol{u}_k^* = [\boldsymbol{u}^*(k|k)^T \ \dots \ \boldsymbol{u}^*(k+N_p-2|k)^T \ \boldsymbol{u}^*(k+N_p-1|k)^T]^T$$

With the injection of the first control action computed, $\boldsymbol{u}^*(k)^T$, the next state will match the predicted state at time $k+1$ since this is the nominal undisturbed case, $\boldsymbol{x}(k+1)$. We show a inherited solution, consisting of the previous optimal solution shifted in one sampling interval and with the state feedback control law delivering the last control action:

$$\tilde{\boldsymbol{u}}_{k+1} = [\boldsymbol{u}^*(k+1|k)^T \ \dots \ \boldsymbol{u}^*(k+N_p-1|k)^T \ [\boldsymbol{u}^*(k+N_p-1|k) - \boldsymbol{K}(\boldsymbol{z}(k+N_p|k) - \boldsymbol{z}_{sp})]^T]^T$$

Problem $\mathscr{P}_7$:

$$\min_{\boldsymbol{u}_k} \quad J(\boldsymbol{x},\boldsymbol{u}), \tag{4.9}$$

where

$$\begin{aligned}
J(\boldsymbol{x},\boldsymbol{u}) &= \left\|\boldsymbol{z}(k+N_p|k)-\boldsymbol{z}_{sp}\right\|_{\boldsymbol{P}}^2 \\
&+ \sum_{j=0}^{N_p-1} \left\|\boldsymbol{z}(k+j|k)-\boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{\Delta u}(k+j|k)\|_{\boldsymbol{R}}^2, && (4.9a)\\
\boldsymbol{\Delta u}(k+j) &= \boldsymbol{u}(k+j|k)-\boldsymbol{u}(k+j-1|k), \quad j\in[0,N_p-1], && (4.9b)\\
\boldsymbol{z}(k+j|k)-\boldsymbol{z}_{sp} &= \begin{bmatrix}\boldsymbol{x}(k+j|k)\\ \boldsymbol{u}(k+j-1|k)\end{bmatrix} - \begin{bmatrix}\boldsymbol{x}_{sp}\\ \boldsymbol{u}_{sp}\end{bmatrix}, \quad j\in[0,N_p-1] && (4.9c)
\end{aligned}$$

subject to

$$\begin{aligned}
&\boldsymbol{x}(k)=\boldsymbol{x}_0, && (4.9d)\\
&\boldsymbol{u}(k-1)=\boldsymbol{u}_0, && (4.9e)\\
&\boldsymbol{x}(k+j+1|k)=\boldsymbol{f}(\boldsymbol{x}(k+j|k),\boldsymbol{u}(k+j|k)), \quad j\in[0,N_p], && (4.9f)\\
&\boldsymbol{x}(k+j|k)\in\mathscr{X}, \quad j\in[0,N_p] && (4.9g)\\
&\boldsymbol{u}(k+j|k)\in\mathscr{U}, \quad j\in[0,N_p-1] && (4.9h)\\
&\boldsymbol{\Delta u}(k+j|k)\in\mathscr{U}_\Delta, \quad j\in[0,N_p-1] && (4.9i)\\
&\left\|\boldsymbol{z}(k+N_p|k)-\boldsymbol{z}_{sp}\right\|_{\boldsymbol{P}}^2 \le \rho, && (4.9j)
\end{aligned}$$

Which is also feasible: the inherited solution up to $k+N_p-1$ was feasible, and by construction the state feedback control law was computed while accounting for state and input constraints. ■

The exponential stability of the closed-loop is given by the following Theorem.

Theorem 3 (Adapted from Theorem 16 of Rajhans et al. (2019)) *Let Assumptions 1-6 hold and Problem $\mathscr{P}_8$ be feasible at k. Then, the nominal undisturbed system with the controller defined in Problem $\mathscr{P}_7$ exponentially stable at the reference value.*

Like most stability proofs, starting point is the optimal solution at time k and its corresponding optimal objective function value.

$$\boldsymbol{u}_k^* = [\boldsymbol{u}^*(k|k)^T \,\dots\, \boldsymbol{u}^*(k+N_p-2|k)^T \; \boldsymbol{u}^*(k+N_p-1|k)^T]^T$$

$$J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k)) = \left\|\boldsymbol{z}^*(k+N_p|k)-\boldsymbol{z}_{sp}\right\|_{\boldsymbol{P}}^2 + \sum_{j=0}^{N_p-1} \left\|\boldsymbol{z}^*(k+j|k)-\boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{\Delta u}^*(k+j|k)\|_{\boldsymbol{R}}^2$$

as well as the inherited solution mentioned in Lemma 3, now accompanied by the corresponding objective function value:

$$\begin{aligned}
\tilde{\boldsymbol{u}}_{k+1} =& [\boldsymbol{u}^*(k+1|k)^T \ \dots \ \boldsymbol{u}^*(k+N_p-1|k)^T \ [\boldsymbol{u}^*(k+N_p-1|k) - \boldsymbol{K}(\boldsymbol{z}(k+N_p|k) - \boldsymbol{z}_{sp})]^T]^T \\
\tilde{J}(\boldsymbol{x}, \tilde{\boldsymbol{u}}) =& \left\|\boldsymbol{z}(k+N_p+1|k) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{P}}^2 + \sum_{j=1}^{N_p} \left\|\boldsymbol{z}(k+j|k) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{\Delta u}(k+j|k)\|_{\boldsymbol{R}}^2 \\
=& \left\|\boldsymbol{z}(k+N_p+1|k) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{P}}^2 + \left\|\boldsymbol{z}(k+N_p|k) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}}^2 + \left\|\boldsymbol{\Delta u}(k+N_p|k)\right\|_{\boldsymbol{R}}^2 \\
&+ \sum_{j=1}^{N_p-1} \left\|\boldsymbol{z}^*(k+j|k) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{\Delta u}^*(k+j|k)\|_{\boldsymbol{R}}^2
\end{aligned}$$

We first make use Lemma 1 which states conditions when the discrete-time LQR stabilizes a nonlinear system. Its condition is that the induced norm $\beta_{\boldsymbol{\Psi}}$ along with its upper bound $\beta_{\boldsymbol{\Psi}}^*$ can be used to determine where in the terminal region Ω the inequality $\varphi(\boldsymbol{z}(k)) \geq 0$ is satisfied, indicating asymptotic stability of the nonlinear system when controlled by the linear quadratic regulator computed by DAREs (3.4) and (3.5). When the induced norm does not violate the upper bound, $\beta_{\boldsymbol{\Psi}} \leq \beta_{\boldsymbol{\Psi}}^*$, the regulator renders the nonlinear system asymptotically stable. We also have the very convenient inequality

$$V_t(\boldsymbol{z}(k+1)) - V_t(\boldsymbol{z}(k)) \leq -\left\|\boldsymbol{z}(k|k) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}_d}^2 = \left\|\boldsymbol{z}(k|k) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}}^2 + \left\|-\boldsymbol{K}(\boldsymbol{z}(k|k) - \boldsymbol{z}_{sp})\right\|_{\boldsymbol{R}}^2 \tag{4.10}$$

which states that the terminal cost is bounded from above by the stage cost of the nonlinear system controlled by the LQR. This inequality can be summed over the time interval $[k, \infty)$, yielding:

$$\left\|\boldsymbol{z}(k|k) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{P}}^2 \geq \sum_{i=k}^{\infty} \left\|\boldsymbol{z}(i) - \boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}_d}^2.$$

The callback to the feedback stabilization of a nonlinear system is relevant due to the relation it guarantees for the comparison of the objective functions $J^*(\boldsymbol{x}(k), \boldsymbol{u}^*(k))$ and $\tilde{J}(\boldsymbol{x}(k), \tilde{\boldsymbol{u}}(k))$. We can write the latter in terms of the former:

$$\begin{aligned}
\tilde{J}(\boldsymbol{x},\tilde{\boldsymbol{u}}) &= \left\|z(k+N_p+1|k)-z_{sp}\right\|_{\boldsymbol{P}}^2 + \left\|z(k+N_p|k)-z_{sp}\right\|_{\boldsymbol{Q}}^2 + \left\|\boldsymbol{\Delta u}(k+N_p|k)\right\|_{\boldsymbol{R}}^2 \\
&+ \sum_{j=1}^{N_p-1} \left\|z^*(k+j|k)-z_{sp}\right\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{\Delta u}^*(k+j|k)\|_{\boldsymbol{R}}^2 \\
&= J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k)) - \left\|z^*(k|k)-z_{sp}\right\|_{\boldsymbol{Q}}^2 - \|\boldsymbol{\Delta u}^*(k|k)\|_{\boldsymbol{R}}^2 \\
&\underbrace{+ \left\|z^*(k+N_p|k)-z_{sp}\right\|_{\boldsymbol{Q}}^2 + \left\|-\boldsymbol{K}(z(k+N_p|k)-z_{sp})\right\|_{\boldsymbol{R}}^2}_{=\left\|z(k+N_p|k)-z_{sp}\right\|_{\boldsymbol{Q}_d}^2} \\
&+ \left\|z(k+N_p+1|k)-z_{sp}\right\|_{\boldsymbol{P}}^2 - \left\|z(k+N_p|k)-z_{sp}\right\|_{\boldsymbol{P}}^2 \\
&= J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k)) - \left\|z^*(k|k)-z_{sp}\right\|_{\boldsymbol{Q}}^2 - \|\boldsymbol{\Delta u}^*(k|k)\|_{\boldsymbol{R}}^2 \\
&+ \left\|z(k+N_p|k)-z_{sp}\right\|_{\boldsymbol{Q}_d}^2 + \left\|z(k+N_p+1|k)-z_{sp}\right\|_{\boldsymbol{P}}^2 - \left\|z(k+N_p|k)-z_{sp}\right\|_{\boldsymbol{P}}^2
\end{aligned}$$

From inequality (4.10), we have that the difference between terminal penalty costs shows an upper bound for the comparison:

$$\begin{aligned}
\tilde{J}(\boldsymbol{x},\tilde{\boldsymbol{u}}) =& J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k)) - \left\|z^*(k)-z_{sp}\right\|_{\boldsymbol{Q}}^2 - \|\boldsymbol{\Delta u}^*(k)\|_{\boldsymbol{R}}^2 \\
&\underbrace{+ \left\|z(k+N_p|k)-z_{sp}\right\|_{\boldsymbol{Q}_d}^2 + \left\|z(k+N_p+1|k)-z_{sp}\right\|_{\boldsymbol{P}}^2 - \left\|z(k+N_p|k)-z_{sp}\right\|_{\boldsymbol{P}}^2}_{\leq 0,\text{ as per manipulation of expression (4.10)}} \\
&\leq J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k)) - \left\|z^*(k|k)-z_{sp}\right\|_{\boldsymbol{Q}}^2 - \|\boldsymbol{\Delta u}^*(k|k)\|_{\boldsymbol{R}}^2
\end{aligned} \tag{4.11}$$

If the optimal solution were to be computed at time $k+1$, the objective function value would be $J^*(\boldsymbol{x}(k+1),\boldsymbol{u}^*(k+1))$ and it can be implied by this optimal objective function value that:

$$J^*(\boldsymbol{x}(k+1),\boldsymbol{u}^*(k+1)) \leq \tilde{J}(\boldsymbol{x}(k+1),\tilde{\boldsymbol{u}}(k+1))$$

Which in turn attends to the inequality (4.11), so that

$$J^*(\boldsymbol{x}(k+1),\boldsymbol{u}^*(k+1)) - J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k)) \leq -\left\|z^*(k|k)-z_{sp}\right\|_{\boldsymbol{Q}}^2 - \|\boldsymbol{\Delta u}^*(k|k)\|_{\boldsymbol{R}}^2 \tag{4.12}$$

With proper selection of $\boldsymbol{Q}$ and $\boldsymbol{R}$ one has that the RHS of the inequality above is negative except when the extended state matches the reference, showing that the optimal objective function value is non-increasing. We now proceed to show exponential stability of the optimal objective function value $J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k))$. We establish this function as a candidate Lyapunov function

$$V(\boldsymbol{x}) = J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k))$$

which is bounded from below according to the following inequality:

$$J^*(\boldsymbol{x}(k),\boldsymbol{u}^*(k)) \geq \lambda_{min}(\boldsymbol{Q})\left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|$$

for all states where the optimization problem is feasible. When the terminal state is in a control invariant set (which the terminal region is, by construction), and

$$\min_{\boldsymbol{u}_k \in \mathscr{U}} \left\{ V_t(\boldsymbol{x}(k+1)) + \left\|\boldsymbol{z}(k|k)-\boldsymbol{z}_{sp}\right\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{\Delta u}(k|k)\|_{\boldsymbol{R}}^2 \right\} \leq V_t(\boldsymbol{x}(k)),$$

then

$$J(\boldsymbol{x}(k),\boldsymbol{u}(k)) \leq V_t(\boldsymbol{x}(k))$$

for every extended state $\boldsymbol{z}(k) \in \Omega$. Then for the objective function value J, we have

$$J(\boldsymbol{x}(k),\boldsymbol{u}(k)) \leq V_t(\boldsymbol{x}(k)) \leq \lambda_{max}(\boldsymbol{P})\left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|^2$$

for extended states in the terminal region Ω. With Proposition 15 of Rajhans et al. (2019), it is stated that if there exists a class K_∞ function $\alpha(\cdot)$ such that

$$J(\boldsymbol{x}(k),\boldsymbol{u}(k)) \leq \alpha(\left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|), \forall \boldsymbol{z}(k) \in \Omega$$

then there is another class K_∞ function $\beta(\cdot)$ such that

$$J(\boldsymbol{x}(k),\boldsymbol{u}(k)) \leq \beta(\left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|), \forall \boldsymbol{z}(k) \in \mathscr{X} \times \mathscr{U},$$

with $\Omega \Subset \mathscr{X}$ and $\mathscr{X}$ a compact set. Then we have a class K_∞ function $\beta(\cdot)$ that bounds $J(\boldsymbol{x}(k),\boldsymbol{u}(k))$ from above. Inequality (4.12) implies that

$$V(\boldsymbol{x}(k+1)) - V(\boldsymbol{x}(k)) < -\eta(\left\|\boldsymbol{z}(k)-\boldsymbol{z}_{sp}\right\|)$$

and for any $0 \leq l \leq k$ we have

$$V(\boldsymbol{x}(k)) - V(\boldsymbol{x}(l)) < -\sum_{i=l}^{k} \eta(\left\|\boldsymbol{z}(i)-\boldsymbol{z}_{sp}\right\|).$$

By consequence, we have that

$$0 < \sum_{i=0}^{\infty} \eta(\left\|\boldsymbol{z}(i)-\boldsymbol{z}_{sp}\right\|) < V(\boldsymbol{x}(0)) - V(\boldsymbol{x}(\infty)) \leq V(\boldsymbol{x}(0)).$$

Finally, upper and lower bounds have been established for the candidate Lyapunov function $V(\boldsymbol{x}(k))$. We also have that the candidate Lyapunov function is zero only when the extended state matches the reference, and that it is strictly positive when not. With the upper and lower bounds for the candidate Lyapunov function class K$_\infty$ functions, the candidate Lyapunov function is in fact a Lyapunov function, pointing to the fact that the closed-loop system with the proposed optimization problem $\mathscr{P}_7$ as its controller asymptotically stable. A further statement can be done regarding exponential stability, when the class K$_\infty$ function $\beta(\cdot)$ is investigated further:

$$\beta(\|z(k)-z_{sp}\|) = \frac{c}{e}\alpha(\|z(k)-z_{sp}\|) = \frac{c}{e}\lambda_{max}(\boldsymbol{P})\left\|z(k)-z_{sp}\right\|^2,$$

with c and e constants that depend on the terminal set Ω and admissible extended states set $\mathscr{X} \times \mathscr{U}$. Then, we have

$$\lambda_{min}(\boldsymbol{Q})\left\|z(k)-z_{sp}\right\|^2 \leq V(\boldsymbol{x}(k)) \leq \frac{c}{e}\lambda_{max}(\boldsymbol{P})\left\|z(k)-z_{sp}\right\|^2 \tag{4.13}$$

and

$$V(\boldsymbol{x}(k+1)) - V(\boldsymbol{x}) < -\lambda_{min}(\boldsymbol{Q})\left\|\boldsymbol{x}(k)\right\|^2 \tag{4.14}$$

showing that in fact, the Lyapunov function obeys the exponential stability criteria. ■

Both the continuous-time and discrete-time cases have drawbacks when it comes to the terminal region calculation since they rely on a solution to the Lyapunov equation, in the work of Chen & Allgöwer (1998), or the DARE equation, in the work of Rajhans et al. (2019). No suggestions as to how to obtain the regulator are made in the former aside from solving the equation for a linearization around the reference, while the latter suggests obtaining the regulator through the *dlqr* function in Matlab for a linearization around the reference. Then both methods proceed to compute the terminal region radius in order to satisfy input constraints and nonlinearity bound constraints through successive optimization problems. Methods for determining these regions were proposed in the previous part, where feedback stabilization is studied for the discrete-time case.

Computation of terminal region and penalty matrix is required at every set point change, since the LQR is stabilizing at the corresponding region computed around an equilibrium point. An offset between the LQR's reference and a set point renders all the theoretical results just shown to be pointless. The LQR cost function used as a terminal weight would no longer be bounded, making it useless as an upper bound of the actual cost function (which is unknown, but known to be bound by the LQR cost function). Also, changes to the LQR's reference would lead to input and/or state constraint violation by the terminal control law. This complicates the use of this formulation with zone control (GONZÁLEZ; ODLOAK, 2009).

In order to improve compatibility between quasi-infinite horizon NMPC and zone control, an additional target calculation layer is proposed. It computes a reference value that yields an equilibrium point that also minimizes an economic term - enabled by zone control. It is represented by the following optimization problem.

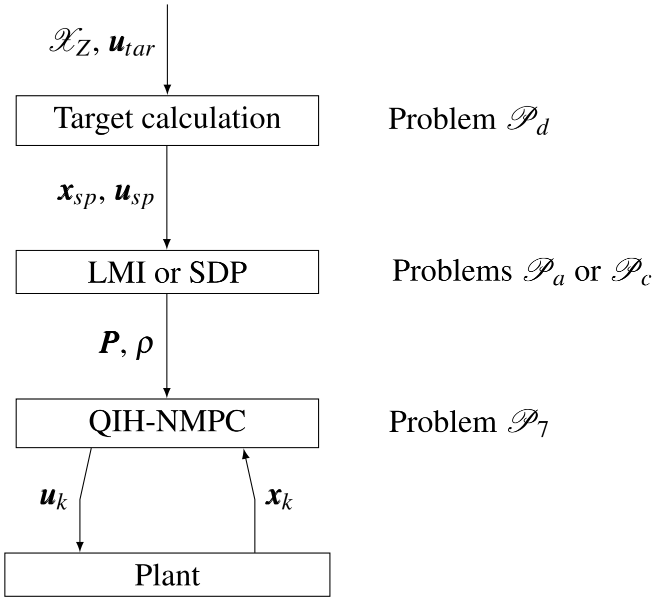
Problem $\mathscr{P}_d$:

$$\min_{\boldsymbol{x}_{sp},\boldsymbol{u}_{sp}} \quad \left\|\boldsymbol{u}_{sp}-\boldsymbol{u}_{tar}\right\|^2_{\boldsymbol{Q}_u}, \tag{4.15}$$

$$\begin{aligned} \text{subject to} \quad & \boldsymbol{f}(\boldsymbol{x}_{sp},\boldsymbol{u}_{sp}) = \boldsymbol{x}_{sp}, & \text{(4.15a)}\\ & \boldsymbol{x}_{sp} \in \mathscr{X}_Z, & \text{(4.15b)}\\ & \boldsymbol{u}_{sp} \in \mathscr{U}, & \text{(4.15c)}\\ & -N\boldsymbol{\Delta u}_{max} \leq \boldsymbol{u}_{sp} - \boldsymbol{u}(k-1|k) \leq N\boldsymbol{\Delta u}_{max}. & \text{(4.15d)} \end{aligned}$$

This optimization is to be executed at every control zone $\mathscr{X}_Z$ change and/or economic target $\boldsymbol{u}_{tar}$ change. Constraint (4.15d) is unnecessary if control move saturation is not considered. This layer computes new reference values $(\boldsymbol{x}_{sp},\boldsymbol{u}_{des})$. These are required to compute a new Jacobian linearization, in order to compute a new terminal region and weight matrix with Problem $\mathscr{P}_a$ or $\mathscr{P}_c$. With zone control, Problem $\mathscr{P}_5$ would only require the solution of either Problems $\mathscr{P}_a$ or $\mathscr{P}_c$, which would send new reference values $(\boldsymbol{x}_{sp},\boldsymbol{u}_{sp})$ to Problem $\mathscr{P}_a$ or $\mathscr{P}_c$, which update $\boldsymbol{P}$ and γ. These values are sent to the NMPC layer, represented by Problem $\mathscr{P}_7$, which computes control actions for the system. This procedure is illustrated by Figure 30.

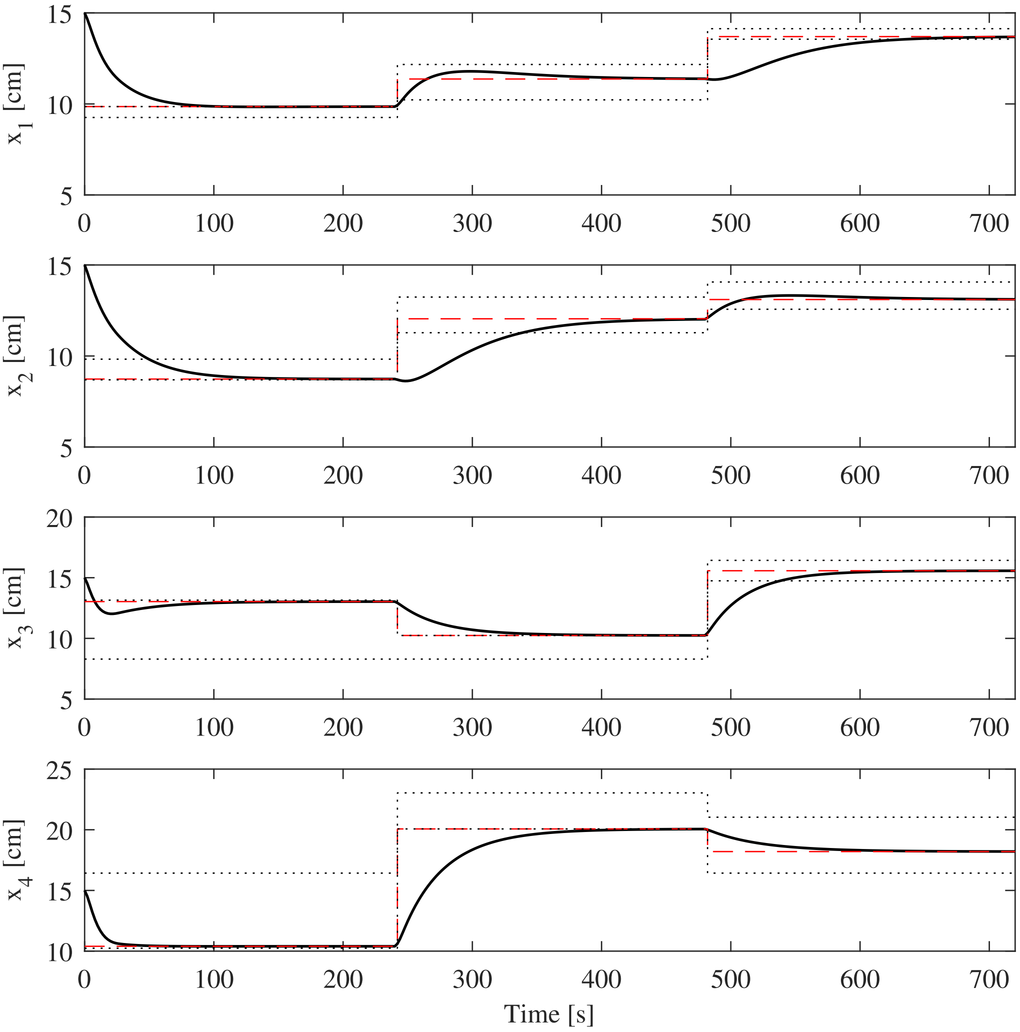
Figure 30 – Block diagram of proposed integration between quasi-infinite horizon NMPC (QIH-NMPC) and zone control



Source: Own author.

We have Figure 31 for the closed-loop states of a simulation performed as per the block diagram of Figure 30. Tuning parameters consist of $\boldsymbol{Q} = 1 \times I_{n_x}$, $\boldsymbol{R} = 1 \times I_{n_u}$, $N_P = N_C = 12$, with sampling time of 2 seconds.

Figure 31 – Zone control with optimizing targets of $\mathscr{P}_7$: states (—), control zones (······) and computed setpoints (- - -)



This formulation tracks the set point computed by the target calculation layer properly. Figure 32, with the inputs and input targets, shows the closed-loop behavior regarding optimizing targets and input move bounds.

Once again, the formulation is capable of bringing inputs to their desired values when said values match the control zones. Also, the input moves were bounded properly as well. The controller cost can be seen in Figure 33.

Once again, as the objective function does not account for the offset between inputs and input targets, the cost converges asymptotically to zero. The computer times and optimization flags can be visualized in Figure 34.

The optimization problem was solved at every sampling interval. The computer times show an interesting behavior: the computational effort increases as the closed-loop system

Figure 32 – Zone control with optimizing targets of $\mathscr{P}_7$: inputs (—) and input targets (- - -)

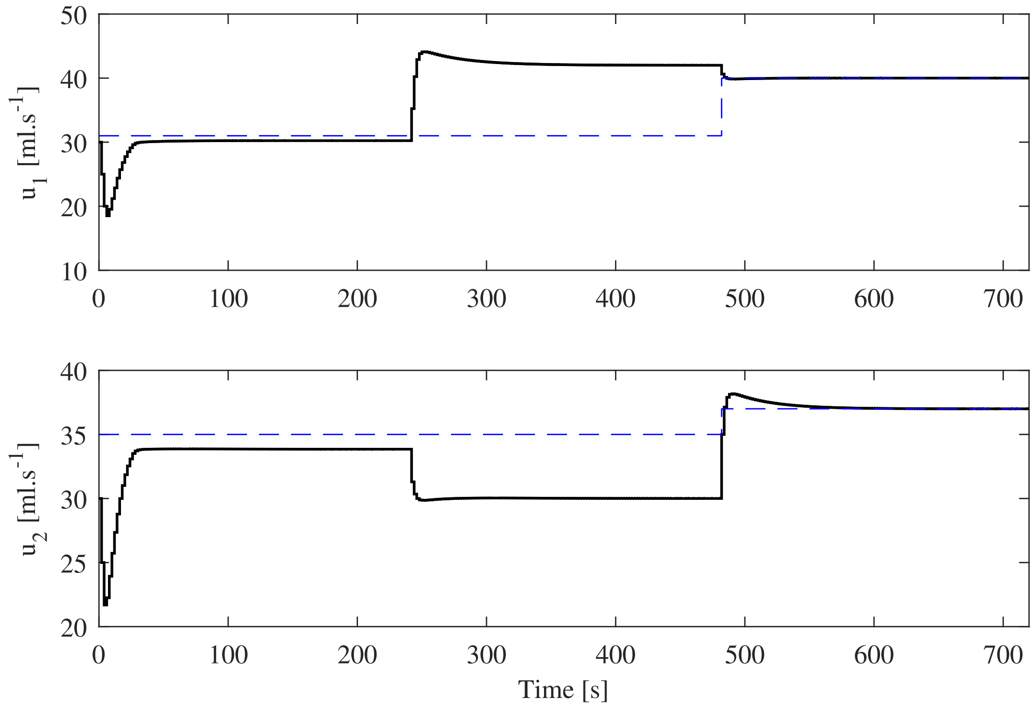
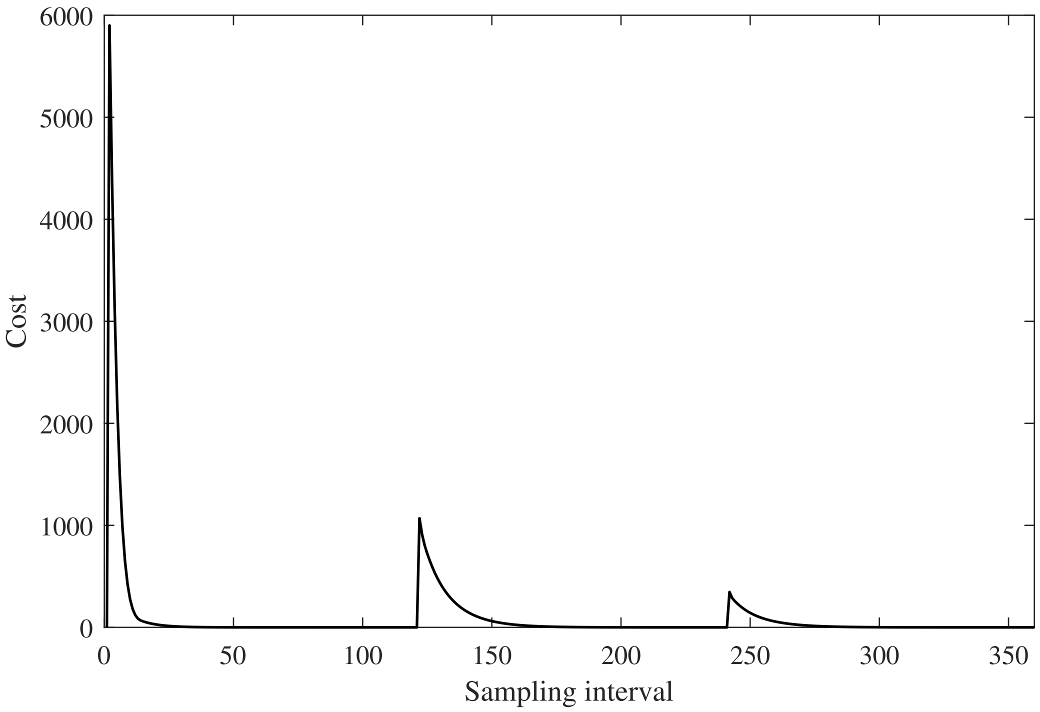


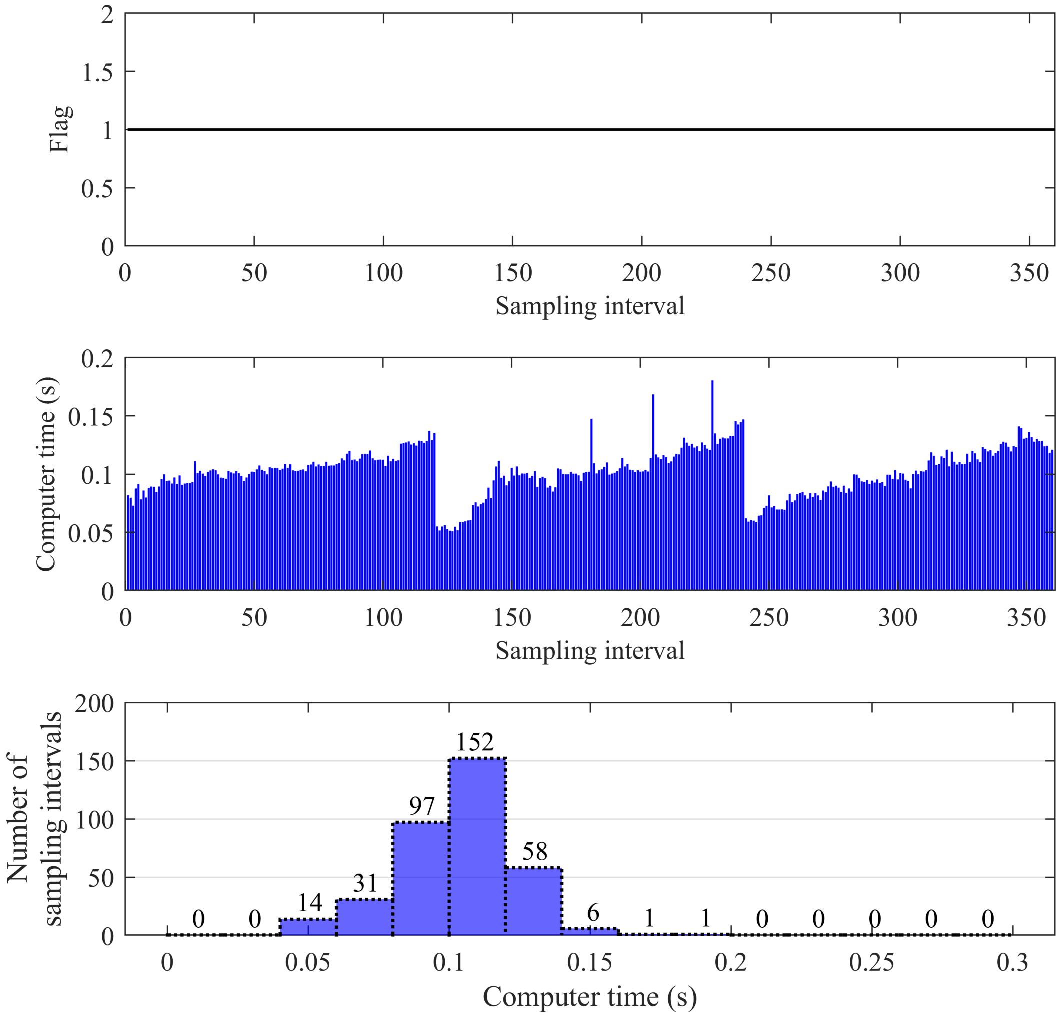
Figure 33 – Zone control with optimizing targets of $\mathscr{P}_7$: cost



converges to the reference. It seems that the inequality constraint causes an increase of the computational effort as the state converges to the reference value.

Regarding computational effort of the computation of the upper layers of target calculation and terminal regions computation in this simulation, computer times for each reference and terminal region determined were respectively $0.4034s$, $0.4692s$, and $0.4231s$. These times are

Figure 34 – Zone control with optimizing targets of $\mathscr{P}_7$: flags and computer times



considerably larger than the computer times used to solve the optimization problem, indicating that the terminal region computation should be performed carefully, and that the choice of computing terminal regions at each change in control zone and/or optimizing target was appropriate. If this computation were to be executed at every sampling interval, it would drastically increase the computational effort required by this formulation.

At the time of this writing, the most recent iteration on the continuous-time case of QIH-NMPC (GRIFFITH; BIEGLER; PATWARDHAN, 2018) proposes changes to the regulator, similar to how it is done by Rajhans et al. (2019), as well as usage of adaptive horizon techniques. Its contribution to the method consists of using multiple open-loop simulations in order to estimate a regulator with a larger domain of attraction than a LQR would have. Although the terminal region is improved, its integration with zone control is not viable since the computational demand to perform these simulations is further increased, which may inviabilize applicability of the controller depending on its sampling time. Also, as has been pointed out in Remark 4 of Chen & Allgöwer (1998), terminal regions for open-loop asymptotically stable systems are unnecessary since the terminal inequality constraint can be removed without loss of stability. However, this has not yet been verified for discrete-time systems, since Chen & Allgöwer (1998) propose a formulation for continous-time systems. This makes the enlargement of the terminal region useful only for cases where this constraint cannot be removed.

4.2.1.3 State contracting constraints

Kothare & Morari (2000) implement a contracting constraint. It enforces contraction of the last state of the prediction horizon with respect to the first state of the prediction horizon. The control policy computed for the whole prediction horizon is then injected into the system, instead of the traditional receding horizon control that injects one control move and computations are performed at the next sampling time. Xie & Fierro (2008) implement a first-state contracting constraint which enforces contraction of the first state of the prediction horizon, yielding a much stricter controller. Sencio et al. (2020) propose a contraction of the error of the last state prediction with respect to the last final state prediction of the previous sampling time. The latter formulation is represented by Problem $\mathscr{P}_9$.

The error of the final state prediction at a sampling time k is represented by $\boldsymbol{\delta}_k$. Terminal state contracting constraint (4.16g) grants Lyapunov stability once $\boldsymbol{\delta}_k$ reaches zero - note that (4.16f) becomes (4.4i) in this situation. Otherwise, an upper bound represented by an exponential decay function has been proven to exist for state trajectories, as is done for other works handling state contracting constraints. This formulation has tuning capabilities due to parameters α_{min} and $\boldsymbol{W}$. Constraining the contracting factor α to values closer to one or small penalization of the distance between α and α_{min} yield to less contracting state trajectories, that is, slower closed-loop dynamics. The stability claim of this formulation is done the following Theorem:

Theorem 4 *Let:*

1. *$\mathscr{X}_Z \cap \mathscr{X}_\omega$ be nonempty;*
2. *$\mathscr{X}$ be an $\mathscr{U}$-controlled invariant set that contains a neighborhood of $\boldsymbol{x}_{sp} \in \mathscr{X}_\omega$;*
3. *there exist a constant $\kappa \in (0,\infty)$ such that $\left\|\boldsymbol{x}(k+j)-\boldsymbol{x}_{sp}\right\| \leq \kappa \left\|\boldsymbol{x}(k)-\boldsymbol{x}_{sp}\right\|$ for any positive integer j.*

Assume Problem $\mathscr{P}_9$ is feasible at time step k, then $\boldsymbol{\delta}_{k+j}$ converges exponentially to the origin as $j \to \infty$.

Item 1 assures that there is a steady state that satisfies constraints (4.16h), (4.16l), and (4.16n). Item 2 further contributes for recursive feasibility of this formulation since it enforces

Problem $\mathscr{P}_8$:

$$\min_{\boldsymbol{u}_k,\boldsymbol{x}_{sp},\boldsymbol{u}_{sp},\boldsymbol{\delta}_k,\alpha_k} \quad J(\boldsymbol{x},\boldsymbol{u}), \tag{4.16}$$

where

$$\begin{aligned} J(\boldsymbol{x},\boldsymbol{u}) = & \sum_{j=0}^{N_p-1} \left\|\boldsymbol{x}(k+j|k)-\boldsymbol{x}_{sp}\right\|_{\boldsymbol{Q}_x}^2 + \sum_{j=0}^{N_c-1} \|\boldsymbol{\Delta u}(k+j|k)\|_{\boldsymbol{R}}^2 \\ & + \left\|\boldsymbol{u}_{sp}-\boldsymbol{u}_{des}\right\|_{\boldsymbol{Q}_u}^2 + \|\alpha_k-\alpha_{min}\|_{\boldsymbol{W}}^2, \end{aligned} \tag{4.16a}$$

subject to

$$\boldsymbol{x}(k|k) = \boldsymbol{x}_0, \tag{4.16b}$$

$$\boldsymbol{u}(k-1|k) = \boldsymbol{u}_0, \tag{4.16c}$$

$$\boldsymbol{x}(k+j+1|k) = \boldsymbol{f}(\boldsymbol{x}(k+j|k),\boldsymbol{u}(k+j|k)), \tag{4.16d}$$

$$\boldsymbol{\Delta u}(k+j|k) = \boldsymbol{u}(k+j|k)-\boldsymbol{u}(k+j-1|k), \tag{4.16e}$$

$$\boldsymbol{\delta}_k = \boldsymbol{x}(k+N_p-1|k)-\boldsymbol{x}_{sp}, \tag{4.16f}$$

$$\|\boldsymbol{\delta}_k\|_S^2 \leq \alpha_k \|\boldsymbol{\delta}_{k-1}\|_S^2, \tag{4.16g}$$

$$\boldsymbol{x}_{sp} = \boldsymbol{f}(\boldsymbol{x}_{sp},\boldsymbol{u}_{sp}), \tag{4.16h}$$

$$\boldsymbol{x}(k+j|k) \in \mathscr{X}, \tag{4.16i}$$

$$\boldsymbol{u}(k+j|k) \in \mathscr{U}, \tag{4.16j}$$

$$\boldsymbol{\Delta u}(k+j|k) \in \mathscr{U}_\Delta, \tag{4.16k}$$

$$\boldsymbol{x}_{sp} \in \mathscr{X}_\omega, \tag{4.16l}$$

$$\boldsymbol{x}_{sp} \in \mathscr{X}_Z, \tag{4.16m}$$

$$\boldsymbol{u}_{sp} \in \mathscr{U}, \tag{4.16n}$$

$$\alpha_k \in [\alpha_{min},1), \tag{4.16o}$$

$$j \in [0,N_p-1] \in \mathbb{N}. \tag{4.16p}$$

the set $\mathscr{X}$, which is not necessarily the state space, be a $\mathscr{U}$-controlled invariant set - in simple terms: the set $\mathscr{U}$ contains all control policies required in order for any state trajectory be set invariant. Item 3 rules out finite escape time systems of the scope of this formulation.

With recursive feasibility, the following relationship between sequential solutions is established:

$$\left\|\boldsymbol{\delta}_{k+j}\right\|_{\boldsymbol{S}}^2 \leq \alpha \left\|\boldsymbol{\delta}_{k+j-1}\right\|_{\boldsymbol{S}}^2 \leq \cdots \leq \prod_{i=1}^{j} \alpha_{k+i} \|\boldsymbol{\delta}_k\|_{\boldsymbol{S}}^2.$$

A maximum value for the contracting factor α is defined as

$$\bar{\alpha} := \max\{\alpha_{k+i} : i \in [1,j], j \in \mathbb{Z}_+\}.$$

The product of all j contracting factors is smaller than the j-th power of the maximum contracting factors, $\bar{\alpha}$,$\prod_{i=1}^{j}\alpha_{k+i} \leq \bar{\alpha}^{j}$. The following inequality that relates the j-th terminal state penalization and the current terminal state penalization:

$$\left\|\boldsymbol{\delta}_{k+j}\right\|_{\boldsymbol{S}}^{2} \leq \bar{\alpha}^{j}\left\|\boldsymbol{\delta}_{k}\right\|_{\boldsymbol{S}}^{2}.$$

With Item 3, there exists a finite number $\kappa > 0$ such that

$$\left\|\boldsymbol{x}(k+N_p-1|k)-\boldsymbol{x}_{sp}\right\|_{\boldsymbol{S}}^{2} \leq \kappa\left\|\boldsymbol{x}(k)-\boldsymbol{x}_{sp}\right\|_{\boldsymbol{S}}^{2}.$$

With the terminal penalty $\boldsymbol{\delta}_k = \boldsymbol{x}(k+N_p-1|k)-\boldsymbol{x}_{sp}$, the expression above multiplied by $\bar{\alpha}^{j}$ results in

$$\bar{\alpha}^{j}\left\|\boldsymbol{\delta}_{k+j}\right\|_{\boldsymbol{S}}^{2} \leq \bar{\alpha}^{j}\kappa\left\|\boldsymbol{x}(k)-\boldsymbol{x}_{sp}\right\|_{\boldsymbol{S}}^{2}.$$

Replacing $\bar{\alpha}^{j}\left\|\boldsymbol{\delta}_{k+j}\right\|_{\boldsymbol{S}}^{2}$ with $\left\|\boldsymbol{\delta}_{k+j}\right\|_{\boldsymbol{S}}^{2}$, the inequality above becomes

$$\left\|\boldsymbol{\delta}_{k+j}\right\|_{\boldsymbol{S}}^{2} \leq \bar{\alpha}^{j}\kappa\left\|\boldsymbol{x}(k)-\boldsymbol{x}_{sp}\right\|_{\boldsymbol{S}}^{2}.$$

With the knowledge that $\bar{\alpha} \leq e^{\bar{\alpha}-1}$, for the j-th power of $\bar{\alpha}$ we have $\bar{\alpha}^{j} \leq e^{-(1-\bar{\alpha})j}$ and

$$\left\|\boldsymbol{\delta}_{k+j}\right\|_{\boldsymbol{S}}^{2} \leq \kappa e^{-(1-\bar{\alpha})j}\left\|\boldsymbol{x}(k)-\boldsymbol{x}_{sp}\right\|_{\boldsymbol{S}}^{2},$$

which means that the norm $\left\|\boldsymbol{\delta}_{k+j}\right\|_{\boldsymbol{S}}^{2}$ has an upper exponential bound, converging exponentially to zero as $k \to \infty$, meaning the terminal state penalty $\boldsymbol{\delta}_{k+j}$ converges to zero as well. ■

As the terminal state penalty converges to zero, the optimization problem becomes the terminal equality constrained formulation, which has had its recursive feasibility and guaranteed stability proven after the optimization problem $\mathscr{P}_6$ was exposed.

Yet, the previous value of the slack variable is not always available, or makes the problem infeasible when there is a control zone change or input target change. The optimization problem $\mathscr{P}_d$ can be used in order to recalculate a new set point and related input to be used in the computation of the previous slack variable that the controller is supposed to contract with the present terminal state prediction $\boldsymbol{x}(k+N_p-1|k)$, as per constraint (4.16g).

This formulation was implemented with the quadruple tank system. Tuning parameters consist of $\boldsymbol{Q}_x = 1 \times I_{n_x}$, $\boldsymbol{Q}_u = 1 \times I_{n_u}$, $\boldsymbol{R} = 1 \times I_{n_u}$, $\boldsymbol{S} = 1 \times I_{n_x}$, $\boldsymbol{W} = 5$, $N_P = 6$, $N_C = 3$, with sampling time of 2 seconds. and the resulting states can be seen in Figure 35.

The controller brings the states to the control zones, computing set points at the border of the zones when input targets are not compatible with control zones. This incompatibility can be confirmed in Figure 36.

Figure 35 – Problem $\mathscr{P}_8$: states (—), control zones (⋯) and computed setpoints (- - -)

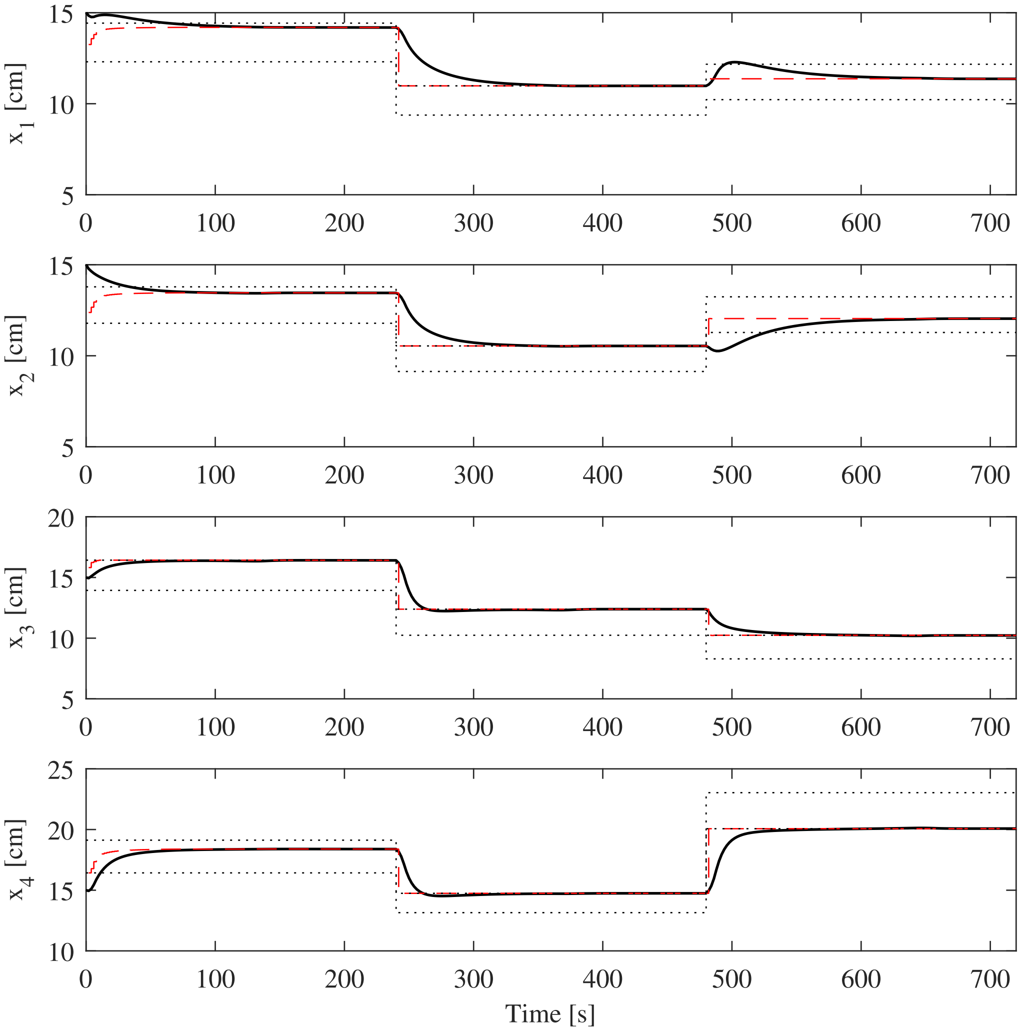
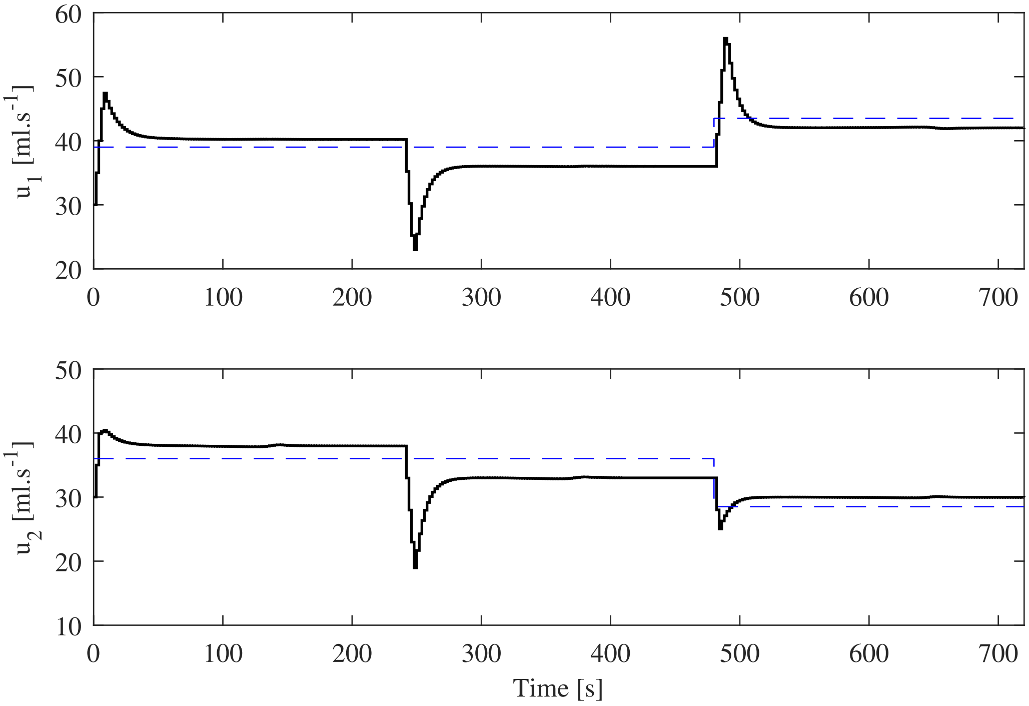
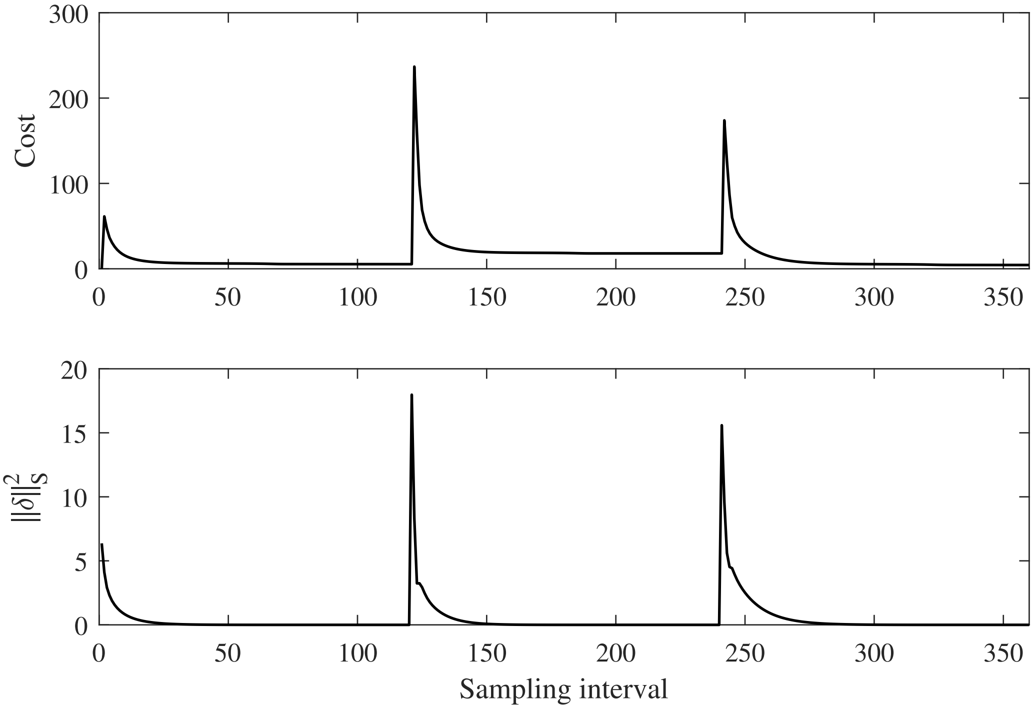


Figure 36 – Problem $\mathscr{P}_8$: inputs (—) and input targets (- - -)



The inputs are brought towards their targets when these are compatible with the control zones, otherwise showing offset. The input move bound is also satisfied. The controller cost and terminal penalty can be seen in Figure 37.

Figure 37 – Problem $\mathscr{P}_8$: cost and terminal penalty



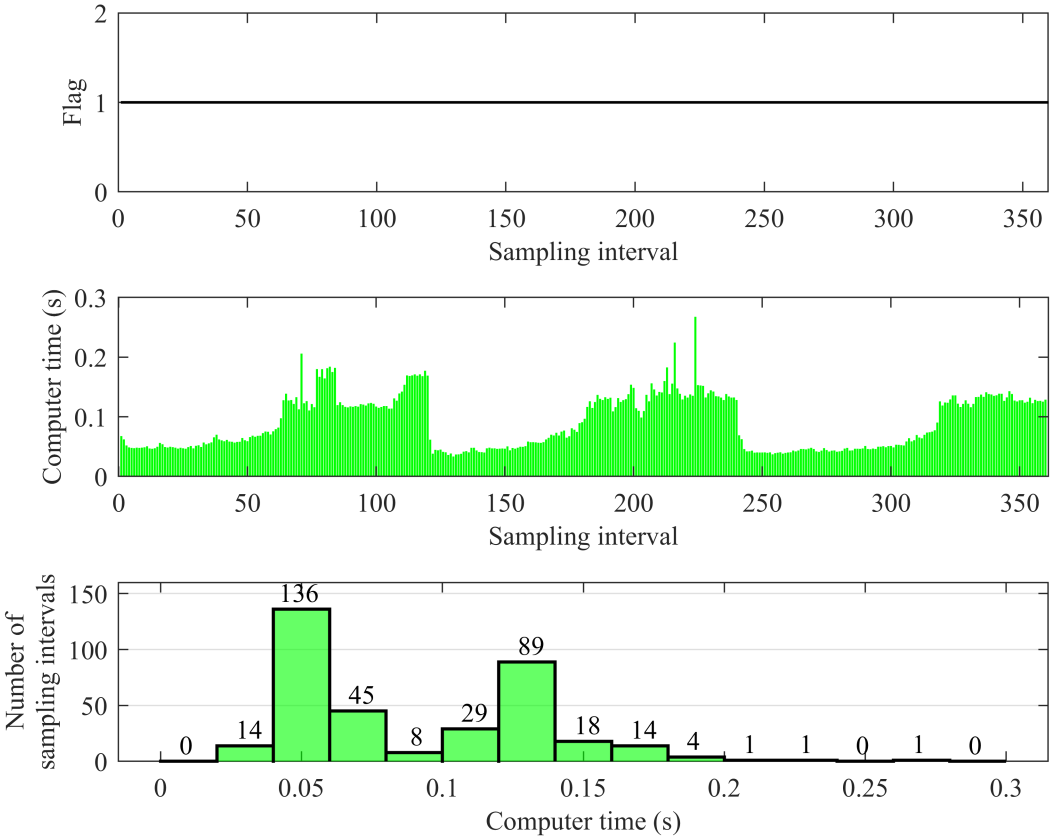
The controller cost should only display non-increasing behavior when the terminal penalty δ is zero. The terminal penalty itself shows non-increasing behavior as it is constrained to do so by the contracting constraint. This is confirmed by Figure 38, which displays the optimization flags as well as the computer times.

The optimization was solved successfully at every sampling interval, and its computer times show an unique behavior, which is related to the contracting parameter α. This is evident in Figure 39.

The optimization has lower solution times when the contraction parameter is not at the bound of its admissible range, which indicates that there might be some solution speed gain if the contraction parameter is less constrained. It is also observed a spike in computational effort in some parts of the simulation, possibly due to the difficulty of computing control actions that contract the terminal cost when the terminal cost is close to zero. Further investigation on this hypothesis is done by simulating once again this closed-loop with an extended admissible range of the contraction parameter. Figure 40 shows such simulation.

This further strengthens the hypothesis that as long as the contracting parameter is not at its admissible bound, obtaining solutions requires less computational effort. This formulation calls for careful penalization of the contracting factor as it might increase the computational effort required to run this formulation online.

Figure 38 – Problem $\mathscr{P}_8$: optimization flags and computer times



4.2.1.4 Other methods

Mejía & Stipanović (2009) propose a cost function contracting constraint. Enforcing controller cost to contract at every sampling interval w.r.t the previous cost forces the controller cost function to be a Lyapunov function. Alamir (2017) proposes a penalty weighting profile on the stage cost instead, also guaranteeing stability as long as such profile exists.

4.3 Computer time improvements

Stability-granting methods have advanced over the years due to continuous search for applicable methods online - stabilizing methods that do not have prohibitive computational effort requirements. But with digitization of industries as a whole, along with evermore powerful and accessible computing power, early and more restrictive methods are accessible once again with the use of advances in optimization algorithms, hardware processing power and numerical methods. Some strategies that may enable earlier stabilizing methods will be discussed briefly in this work as they will be used later.

Figure 39 – Problem $\mathscr{P}_8$: computer times and contracting parameter

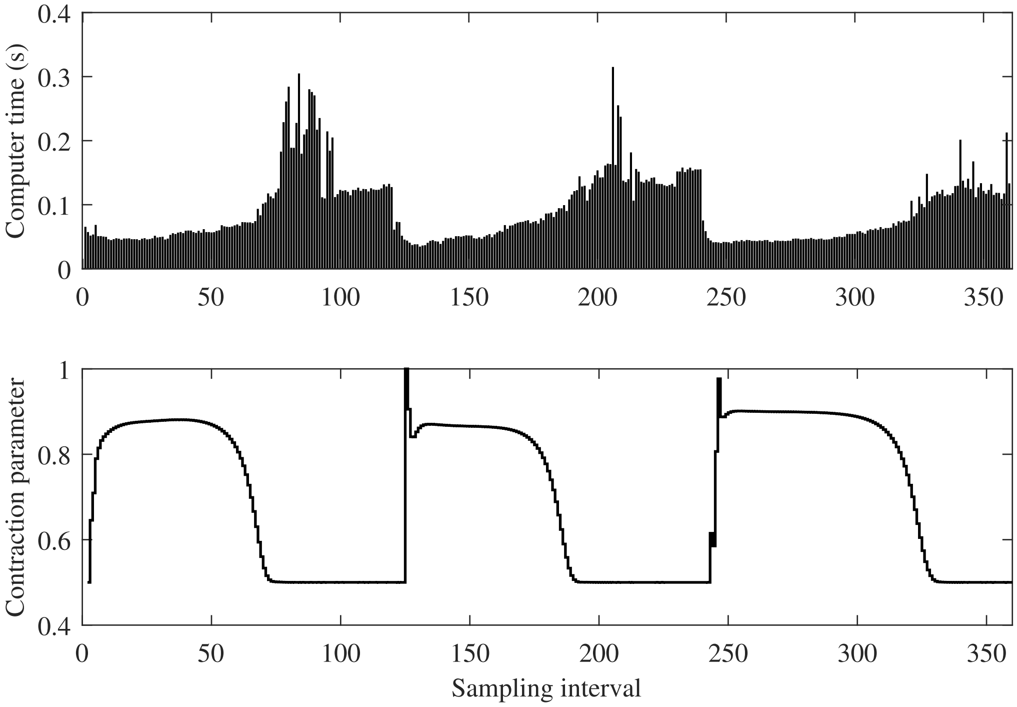
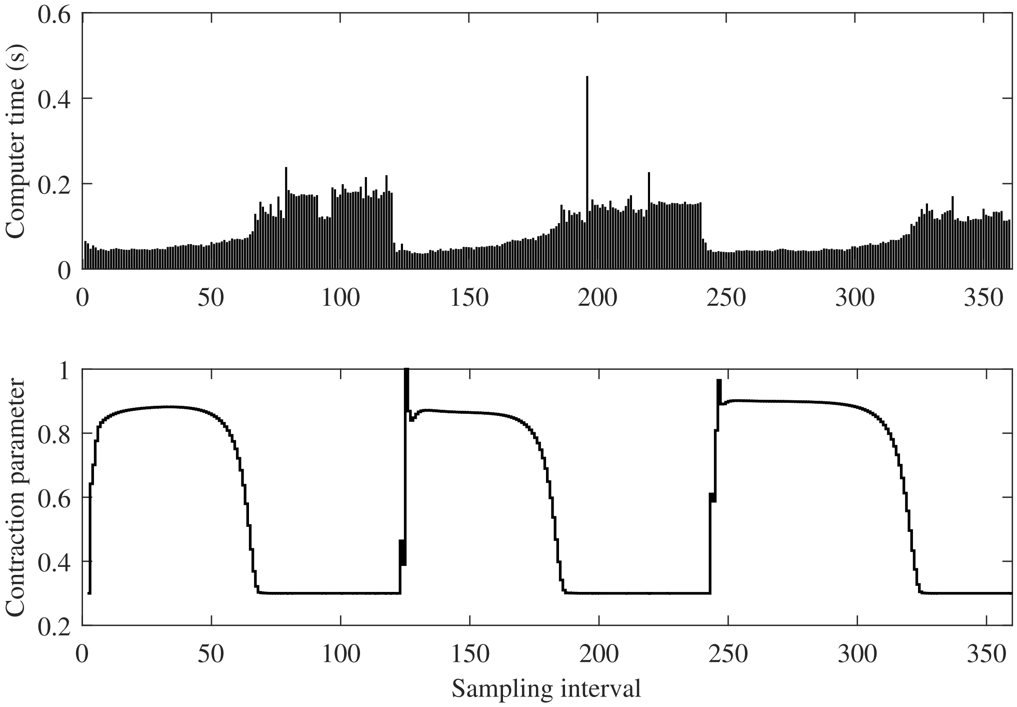


Figure 40 – Problem $\mathscr{P}_8$: computer times and a less constrained contracting parameter

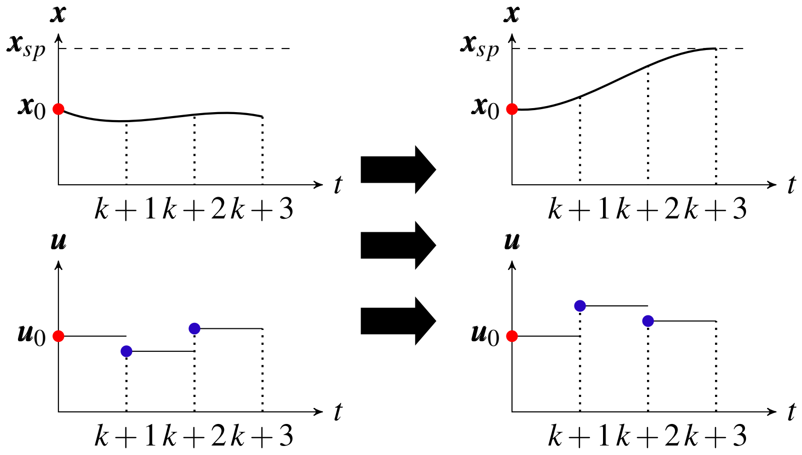


4.3.1 Shooting methods

Different shooting methods consist of how state trajectories are to be handled by the optimization algorithm.

Single shooting method consists of having the state trajectory prediction as a constraint while input is a degree of freedom for each sampling interval. A numerical integrator is the relation between the degree of freedom (input) with the constraint (state prediction). This yields one constraint and one degree of freedom for every sampling interval over the controller horizon. This can be visualized in Figure 41.

Figure 41 – Exemplary single shooting optimization with degrees of freedom (●) and a single, fixed shooting point (●)

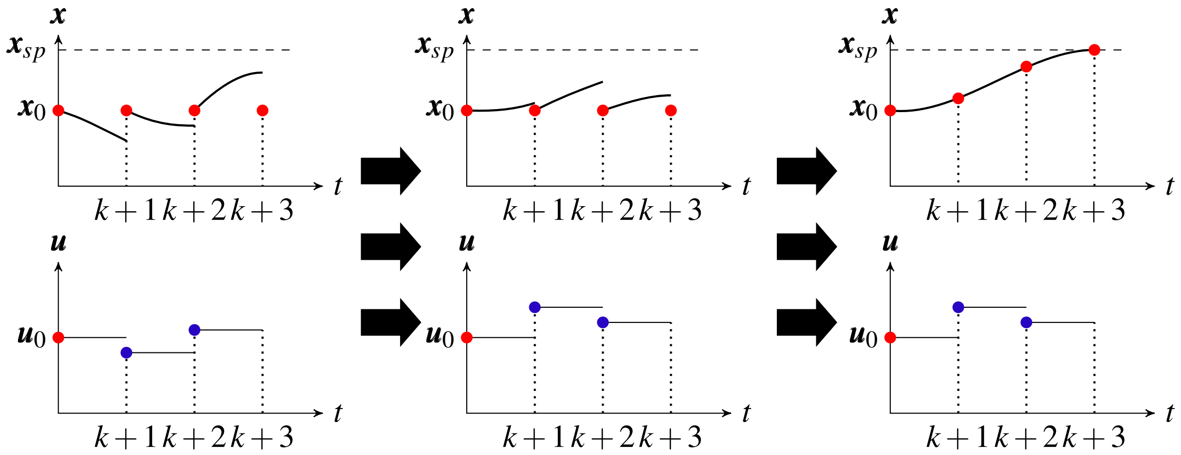


Source: Own author

Figure 41 shows in detail how the single shooting point is the starting point where the optimization algorithm will seek for an optimal control policy by optimizing the input values that are degrees of freedom.

Multiple shooting method differs in how the state trajectory prediction is handled. Inputs over the control horizon are degrees of freedom along with state appending points for every sampling interval. Constraint-wise, there are two, one that represents the integrator and another that appends the state at the end of one sampling interval with the state at the start of the next sampling interval. This is illustrated in Figure 42.

Figure 42 – Exemplary multiple shooting optimization with degrees of freedom (●) and multiple shooting points (●)



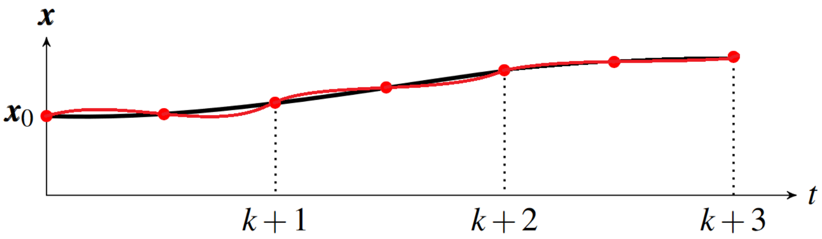
Source: Own author

Figure 42 shows multiple shooting points - the first, fixed as in single shooting, and the others, degrees of freedom. The optimization is implemented with an additional constraint for every sampling interval which appends the state at the end of one sampling interval with the state at the start of the next sampling interval. The additional degrees of freedom help with convergence speed since there are multiple less constrained state trajectories instead of a single trajectory. In the literature, multiple shooting is referred as the transformation of a single boundary value problem into multiple, smaller initial value problems.

4.3.2 Numerical integration

Orthogonal polynomial collocation replaces the numerical integration step with collocation of an orthogonal polynomial over specific appending points, determined via Lagrange interpolation polynomials - the number of appending points is the order of the Lagrange polynomial. The quadrature, as this procedure is commonly referred to in the literature, is significantly faster than a numerical integrator. The number of appending points determines the method's numeric precision in representing the state trajectory. Furthermore, the selection of the quadrature weights also increases the quadrature's accuracy. There are several methods of weight calculation, with the most traditional being Gauss-Jacobi, Gauss-Radau and Gauss-Lobatto (RICE; DO, 2012). Figure 43 illustrates how a collocation orthogonal polynomial would behave along with a state trajectory calculated by a numerical integrator.

Figure 43 – Orthogonal polynomial (🟥) collocated over $N = 3$ appending points (●) and numerical integration step of a state trajectory (—)



Source: Own author.

It is evident that the method is most precise at the appending points. For the purpose of NMPC, any quadrature method with appropriate accuracy should yield faster computer times and appropriate state predictions when compared to an optimization problem that relies on numerical integrators in order to make state predictions.

5 Closed-loop simulations

In this chapter, there are two systems that were simulated in closed-loop, the jacketed CSTR - already exposed in the case study in Subsection 2.2.1 - and the traditional quadruple-tank system benchmark, which will be exposed later in this subsection.

5.1 Jacketed CSTR with finite horizon nonlinear model predictive controller

A closed-loop simulation of the jacketed CSTR system from the case study in Subsection 2.2.1 was performed with a nominal finite horizon nonlinear model predictive controller designed for zone tracking (GONZÁLEZ; ODLOAK, 2009), represented by Problem $\mathscr{P}_5$.

Controller parameters selected have followed no previous criteria and were mostly selected as identity matrices. They are

$$\boldsymbol{Q}_x = \text{diag}(1 \quad 1 \quad 1),\ \boldsymbol{R} = \text{diag}(1 \quad 1),\ \boldsymbol{Q}_u = \text{diag}(1 \quad 1),$$
$$N_p = N_c = N = 10,\ t_{sampling} = 10\text{min}.$$

The closed-loop simulation is divided in four parts. It starts with the reactor states at a start-up condition, $\boldsymbol{x}_0$, with starting input $\boldsymbol{u}_0$. The control zones the closed-loop is to traverse consist of a high conversion state named ignition zone $\mathscr{X}_{ignition}$, a low conversion state called extinction zone $\mathscr{X}_{extinction}$, and an intermediary conversion named unstable zone $\mathscr{X}_{unstable}$. All of these zones have the same economic target $\boldsymbol{u}_{target,1}$ until the fourth part where the economic target becomes $\boldsymbol{u}_{target,2}$.

$$
\begin{aligned}
\boldsymbol{x}_0 = \begin{bmatrix}1 & 0 & -1\end{bmatrix}^T, \boldsymbol{u}_0 = \begin{bmatrix}0.9 & 0.75\end{bmatrix}^T,\\
\mathscr{X} = \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix}0 & -1 & -1\end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix}1 & 8 & 8\end{bmatrix}^T\right\},\\
\mathscr{U} = \left\{\boldsymbol{u} \in \mathbb{R}^{n_u} \middle| \begin{bmatrix}0 & 0\end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix}5 & 10\end{bmatrix}^T\right\},\\
\mathscr{U}_\Delta = \left\{\boldsymbol{\Delta u} \in \mathbb{R}^{n_u} \middle| \begin{bmatrix}-0.4 & -0.8\end{bmatrix}^T \leq \boldsymbol{\Delta u} \leq \begin{bmatrix}0.4 & 0.8\end{bmatrix}^T\right\},\\
\mathscr{X}_{ignition} = \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix}0 & 4 & -1\end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix}0.4 & 8 & 8\end{bmatrix}^T\right\},\\
\mathscr{X}_{extinction} = \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix}0.8 & -1 & -1\end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix}1 & 1 & 8\end{bmatrix}^T\right\},\\
\mathscr{X}_{unstable} = \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix}0.5 & 2 & -1\end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix}0.6 & 4 & 8\end{bmatrix}^T\right\},\\
\boldsymbol{u}_{target,1} = \begin{bmatrix}1 & 1\end{bmatrix}^T, \boldsymbol{u}_{target,2} = \begin{bmatrix}1 & 0\end{bmatrix}^T.
\end{aligned}
$$

The controller must navigate the states from zone to zone while maintaining nominal inputs represented by the first input target. The economic target change was deployed at the unstable zone to evaluate its stabilizing properties in the unstable zone. The resulting closed-loop states can be seen in Figure 44.

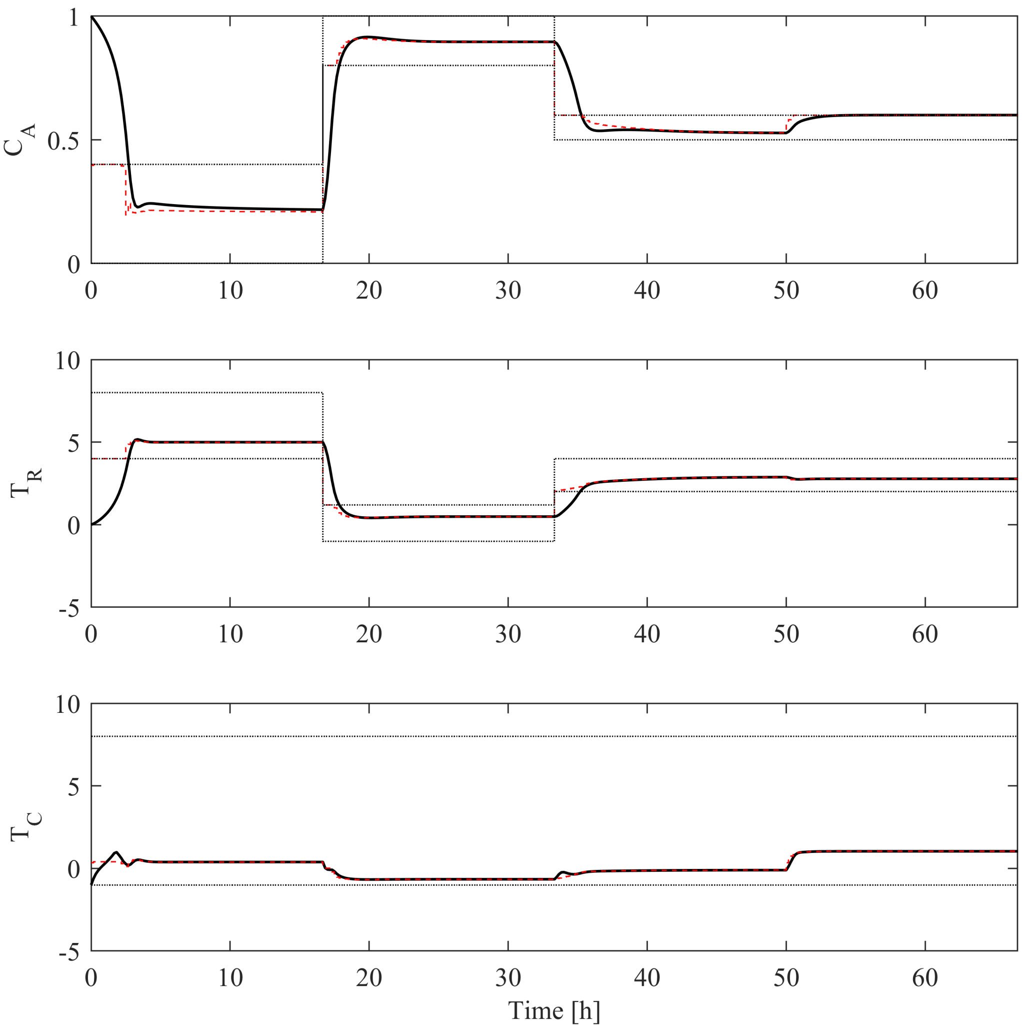
The controller is capable of tracking these zones satisfactorily, with the additional information that the change in input target brings the closed-loop states to the zone frontier, but not violating the zone control condition. The third state was specifically left unconstrained zone-wise in order to have a safer operating reactor. The computed inputs are observed in Figure 45.

The inputs are computed in such a manner that both the actuator saturation as well as the input saturation constraints are not violated. Also, as the states are at the zone frontier, inputs show offset w.r.t. input target. However, the inputs show rather drastic overshoot and sluggish responses overall, which is not necessarily connected to penalty weight tuning. The controller cost can be visualized in Figure 46.

The cost function shows a non-increasing behavior, which may indicate closed-loop stability without any stability-granting element. There are no constraints that grant this controller stability in the sense of Lyapunov, but this formulation's closed-loop asymptotic stability is related to how close the FHOCP is from representing an IHOCP. Although the cost function seems to be a Lyapunov function, the closed-loop performance is yet to be evaluated.

Grüne (2012) relates the length of the FHOCP horizon with a suboptimality index α_O, which is a metric of how far the FHOCP is from representing the full IHOCP. Grüne & Pannek (2011) go in detail how to compute this index online (*a posteriori*) or offline (*a priori*), with the

Figure 44 – Jacketed CSTR controlled by Problem $\mathscr{P}_5$: states (—), control zones (⋯⋯) and computed setpoints (- - -)



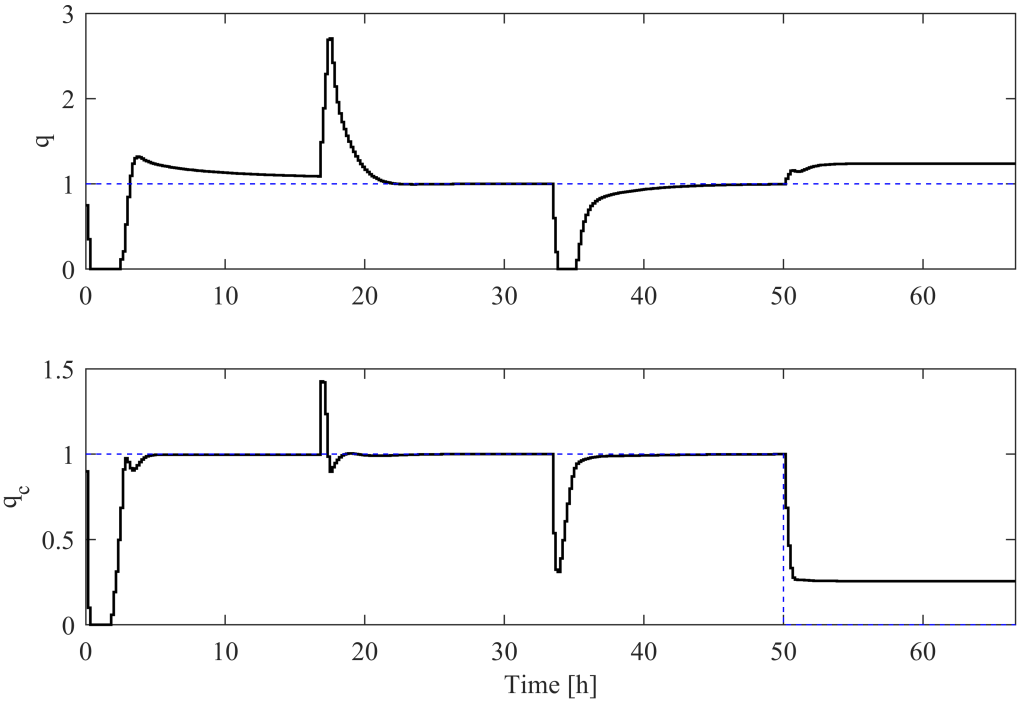
Source: Own author.

latter being a better estimate although it has a larger computational effort than the former. The online estimate is done through the following equation:

$$\alpha_{O,k} = \frac{V_{k-1} - V_k}{l(k-1)}. \tag{5.1}$$

The suboptimality index is a real number in the $[0,1]$ interval is useful in online diagnostics of closed-loop systems. It is a measure of how close the closed-loop is from an IHOCP by comparing the stage cost from the previous sampling, $l(k-1)$ with the difference between the previous controller cost, V_{k-1}, and the current controller cost, V_k. This ratio is based on the Dynamic Programming principle, which states that an optimal problem can be partitioned into multiple optimal sub problems. This applies to NMPC by comparison of the current FHOCP

Figure 45 – Jacketed CSTR controlled by Problem $\mathscr{P}_5$: inputs (—) and input targets (- - -)



Source: Own author.

solution w.r.t. the FHOCP solution from the previous sampling time.

This index is not applicable to controllers with a terminal penalty term. Transforming the economic term of the objective function into a stage cost component does not resolve this issue alone. Consider a closed-loop in equilibrium. The change in controller cost from the last to the current sampling time would be:

$$V_{k-1} - V_k = 0,$$

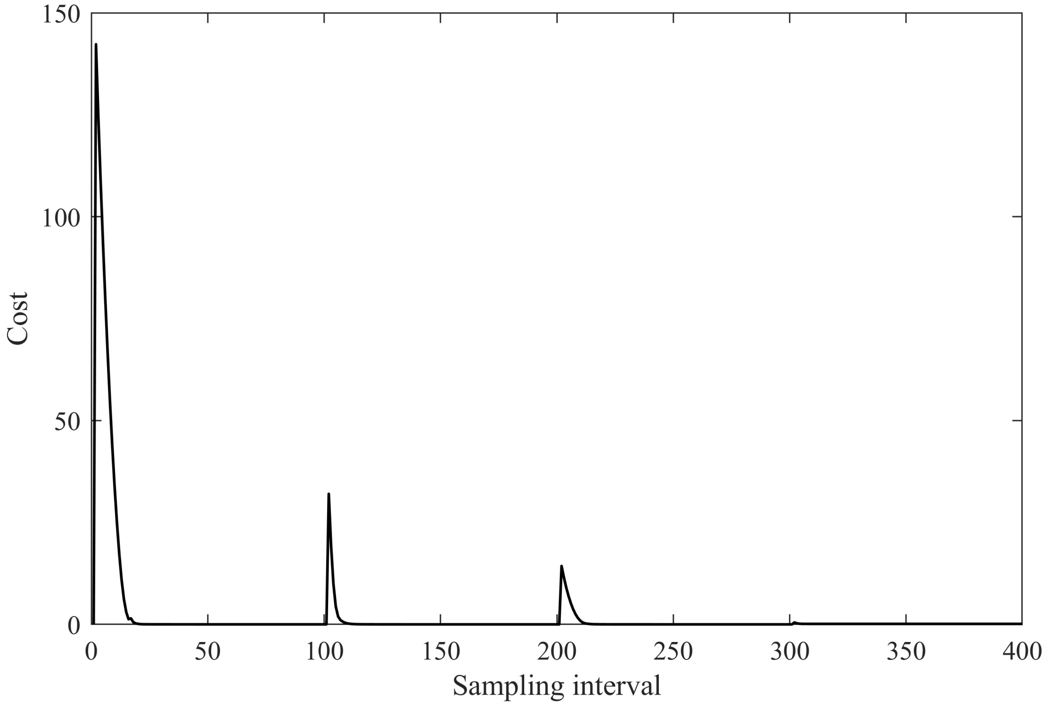
where the stage cost would be

$$l(k-1) = \left\|\boldsymbol{x}(k-1|k-1) - \boldsymbol{x}_{sp,k-1}\right\|^2_{\boldsymbol{Q}_x} + \|\boldsymbol{\Delta u}(k-1|k-1)\|^2_{\boldsymbol{R}} + \|\boldsymbol{u}(k-1|k-1) - \boldsymbol{u}_{des}\|^2_{\boldsymbol{Q}_u}.$$

The online suboptimality index $\alpha_{O,k}$ would then be the ratio of a nominator which tends to zero and a denominator which is not null unless the input target $\boldsymbol{u}_{des}$ is the corresponding input of the set point $\boldsymbol{x}_{sp,k-1}$.

However, the approximation of a IHOCP by extension of a FHOCP's horizon could yield similar performance, if not better, to a controller with stabilizing elements such as the formulations discussed in Section 4.2.1. Multiple closed-loop simulations were performed with different horizon lengths in order to evaluate what performance can be expected from the closed-

Figure 46 – Jacketed CSTR controller cost



Source: Own author.

loop system. The resulting closed-loop states of various controller horizon lengths can be seen in Figure 47.

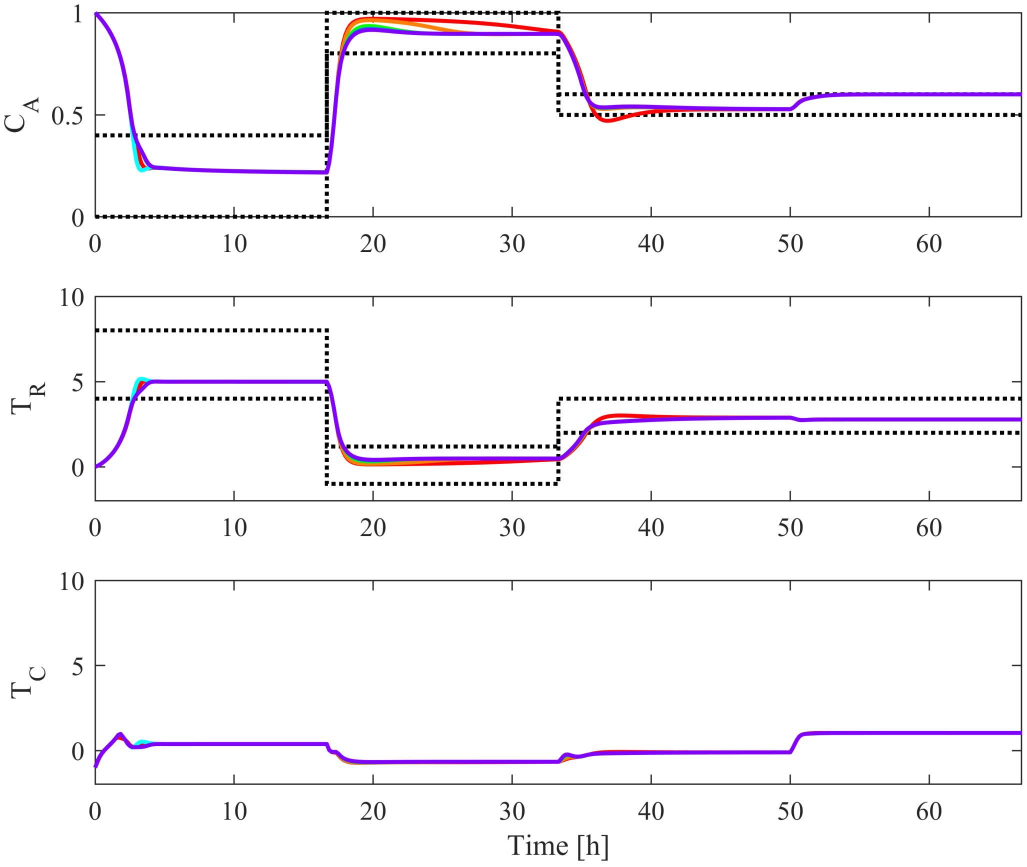
From the states perspective, the closed-loop seems to have a substantial performance gain up until $N = 8$. Gradual increases to controller horizon show that a FHOCP can have very similar performances even with drastically larger horizons. Tuning this FHOCP horizon can yield an optimization with smaller scale but near-optimal performance. The computed inputs for the same varied controller horizon lengths can be seen in Figure 48.

For computed inputs, once again the closed-loop shows substantial performance gains up until N=8. It can also be seen that the sluggish response seen in the first closed-loop simulations has been resolved with the controller horizon extension. Finally, the controller cost for multiple controller horizon lengths can be seen in Figure 49.

The controller cost increase is to be expected as one increases the controller horizon, but it can be seen that there is a limit to how much the cost increases. This upper bound is the optimal cost of an IHOCP, the optimization problem that encompasses all of the system dynamics over time. There is a diminishing return to horizon length increases, especially for systems with short sampling time. This is better seen in Figure 50.

In detail, the cost function converges to the optimal cost at horizon length N=25. Beyond this length, the controller cost is the same throughout the entire simulation, showing the

Figure 47 – Jacketed CSTR state trajectory under controller $\mathscr{P}_6$ with multiple horizon lengths: N=3 (—), N=5 (—), N=8 (—), N=10 (—), N=25 (—), N=50 (—)



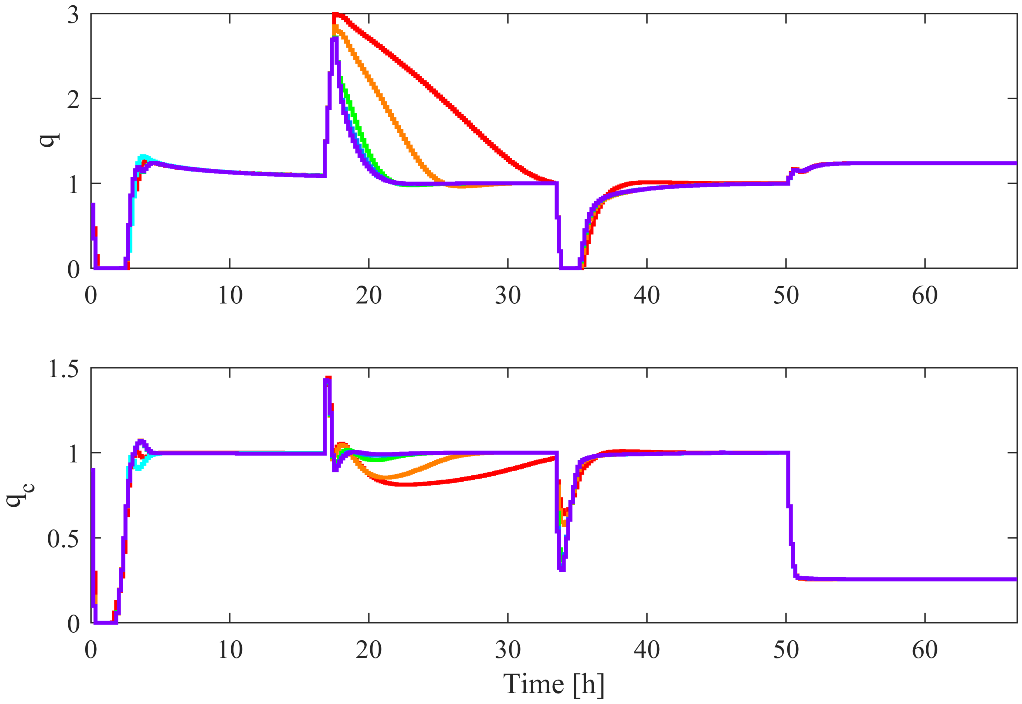
Source: Own author.

diminishing return in excessively enlarging the horizon length.

Horizon tuning for this type of controller should take into account both the performance losses a short horizon would bring to the controller and the increased computational effort related to a longer horizon controller. With this in mind, the practitioner should consider the sum of the controller cost as indicative of convergence of the controller cost to the IHOCP cost, or variations of said sum when compared to sums produced by the same controller but with different horizon lengths.

Now that the tuning process of a controller without guaranteed stability has been exposed, the controllers with stabilizing elements and their corresponding simulations can be shown.

Figure 48 – Jacketed CSTR inputs computed by controller $\mathscr{P}_6$ with multiple horizon lengths: N=3 (—), N=5 (—), N=8 (—), N=10 (—), N=25 (—), N=50 (—)



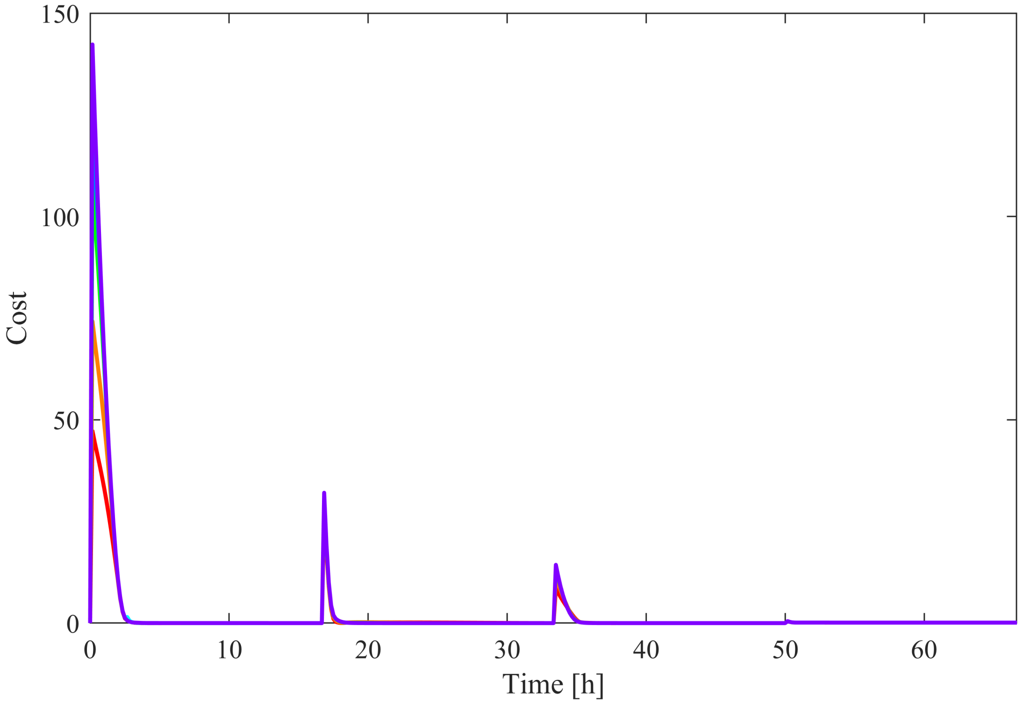
Source: Own author.

5.2 Quadruple-tank system with nonlinear model predictive controllers

In this subsection, the stabilizing formulations described in Section 4.2.1 will be deployed in closed-loop with a plant. As is known, the quasi-infinite horizon NMPC depends on a stabilizable Jacobian linearization of the plant around the desired reference value, the unstable reactor would be an unfair benchmark to evaluate these different stabilizing elements.

For this system, the closed-loop simulation is divided in three parts.The starting state $\boldsymbol{x}_0$ and input $\boldsymbol{u}_0$ are then to traverse three control zones ($\mathscr{X}_{z,1}$, $\mathscr{X}_{z,2}$, $\mathscr{X}_{z,3}$) with an input target change ($\boldsymbol{u}_{target,1}$ to $\boldsymbol{u}_{target,2}$) in the third part. Of course, the states, inputs and input moves are constrained as well by their respective sets $\mathscr{X}$, $\mathscr{U}$, and $\mathscr{U}_\Delta$. The starting state and input values, control zones the formulations are to traverse, and the input targets the formulations must attend are as follows:

Figure 49 – Cost function of controller $\mathscr{P}_6$ with multiple horizon lengths: N=3 (—), N=5 (—), N=8 (—), N=10 (—), N=25 (—), N=50 (—)



Source: Own author.

$$\boldsymbol{x}_0 = \begin{bmatrix} 15 & 15 & 15 & 15 \end{bmatrix}^T, \boldsymbol{u}_0 = \begin{bmatrix} 30 & 30 \end{bmatrix}^T,$$

$$\mathscr{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix} 7.5 & 7.5 & 3.5 & 4.5 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 28 & 28 & 28 & 28 \end{bmatrix}^T \right\},$$

$$\mathscr{U} = \left\{ \boldsymbol{u} \in \mathbb{R}^{n_u} \middle| \begin{bmatrix} 0 & 0 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 60 & 60 \end{bmatrix}^T \right\},$$

$$\mathscr{U}_\Delta = \left\{ \boldsymbol{\Delta u} \in \mathbb{R}^{n_u} \middle| \begin{bmatrix} -5 & -5 \end{bmatrix}^T \leq \boldsymbol{\Delta u} \leq \begin{bmatrix} 5 & 5 \end{bmatrix}^T \right\},$$

$$\mathscr{X}_{z,1} = \left\{ \boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix} 12.3 & 11.8 & 13.9 & 16.4 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 14.4 & 13.8 & 16.4 & 19.1 \end{bmatrix}^T \right\},$$

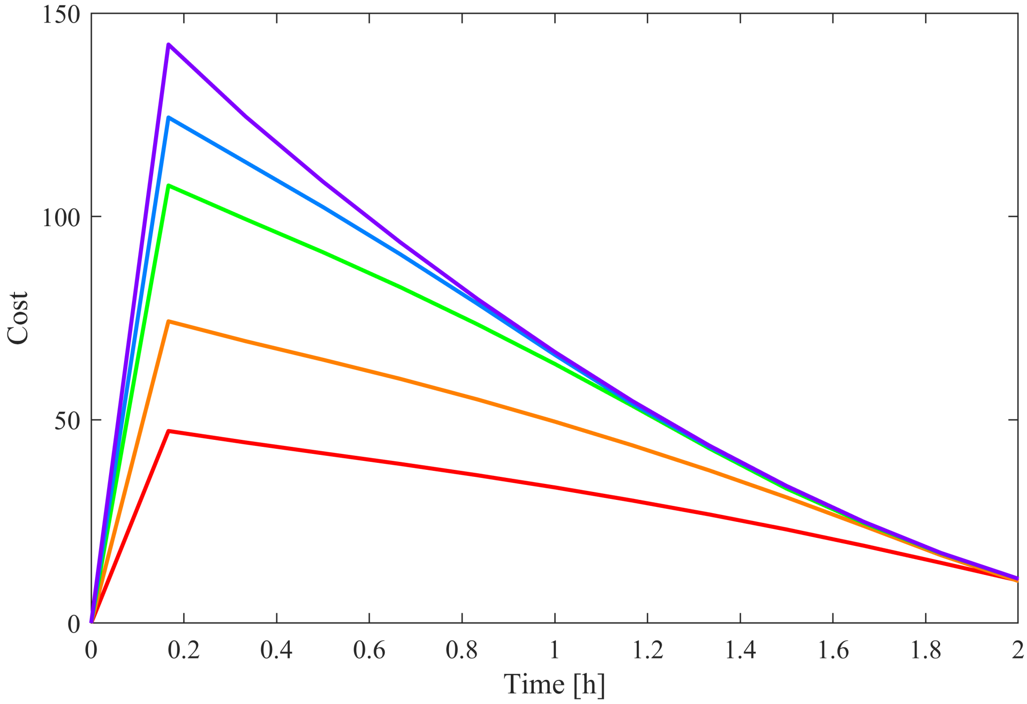
$$\mathscr{X}_{z,2} = \left\{ \boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix} 9.4 & 9.1 & 10.2 & 13.2 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 11.0 & 10.5 & 12.4 & 14.7 \end{bmatrix}^T \right\},$$

$$\mathscr{X}_{z,3} = \left\{ \boldsymbol{x} \in \mathbb{R}^{n_x} \middle| \begin{bmatrix} 10.2 & 11.3 & 8.3 & 20.1 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 12.2 & 13.2 & 10.2 & 23.0 \end{bmatrix}^T \right\},$$

$$\boldsymbol{u}_{target,1} = \begin{bmatrix} 39 & 36 \end{bmatrix}^T, \boldsymbol{u}_{target,2} = \begin{bmatrix} 43.5 & 28.5 \end{bmatrix}^T.$$

As multiple controllers were deployed, their parameters are:

Figure 50 – Zoom of cost function of controller $\mathscr{P}_6$ with multiple horizon lengths: N=3 (—), N=5 (—), N=8 (—), N=10 (—), N=25 (—), N=50 (—)



Source: Own author.

$$\boldsymbol{Q}_x = \text{diag}(1 \quad 1 \quad 1 \quad 1),\ \boldsymbol{R} = \text{diag}(1 \quad 1),\ \boldsymbol{Q}_u = \text{diag}(1 \quad 1),$$

$$t_{sampling} = 2\text{s},\ t_{sim} = 360 t_{sampling},$$

$$\alpha_{min} = 0.5,\ \boldsymbol{S} = \text{diag}(1 \quad 1 \quad 1 \quad 1),\ W = 100.$$

Their prediction and control horizons can be visualized in Table 1.

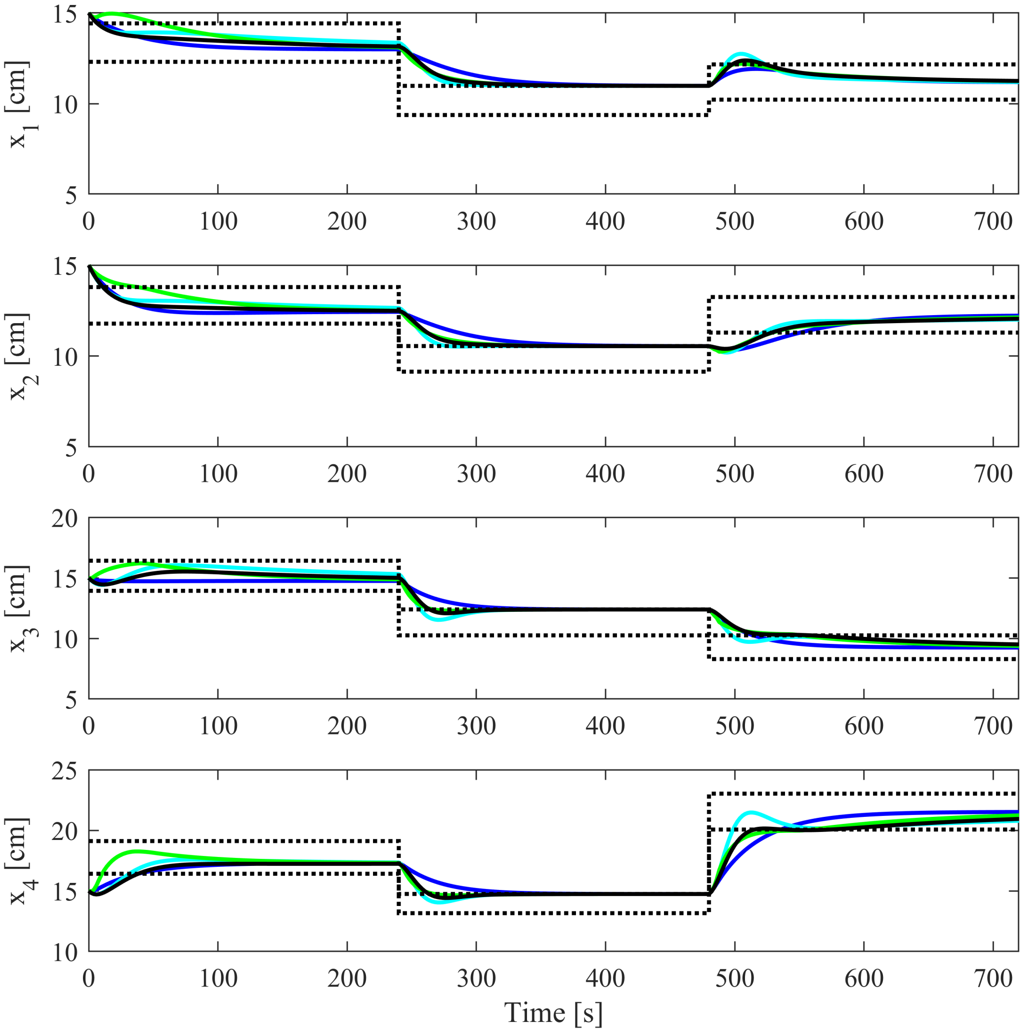
Table 1 – Table of prediciton (N_p) and control (N_c) horizons of NMPC formulations

Terminal constraint	N_p	N_c
Equality (TEC)	27	27
Inequality (TIC)	12	12
Contracting (TCC)	8	4
None (NC)	18	18

Source: Own author.

Terminal equality constraints require larger horizons since the equality constraint is to be satisfied. Since terminal inequality and contracting constraints are more relaxed, they are feasible with shorter horizons. As has been shown for the unstable reactor, short horizons lead to poor performance of NMPCs without guaranteed stability. By trial and error, the no guaranteed stability formulation horizon was increased up until the objective function did not show any significant change in its value over simulations while increasing horizon lengths. The resulting closed-loop state trajectories is seen in Figure 51.

Figure 51 – Quadruple-tank system state trajectory for different controllers: NC (—), TCC (—), TEC (—), and TIC (—) formulations for control zones (·····)

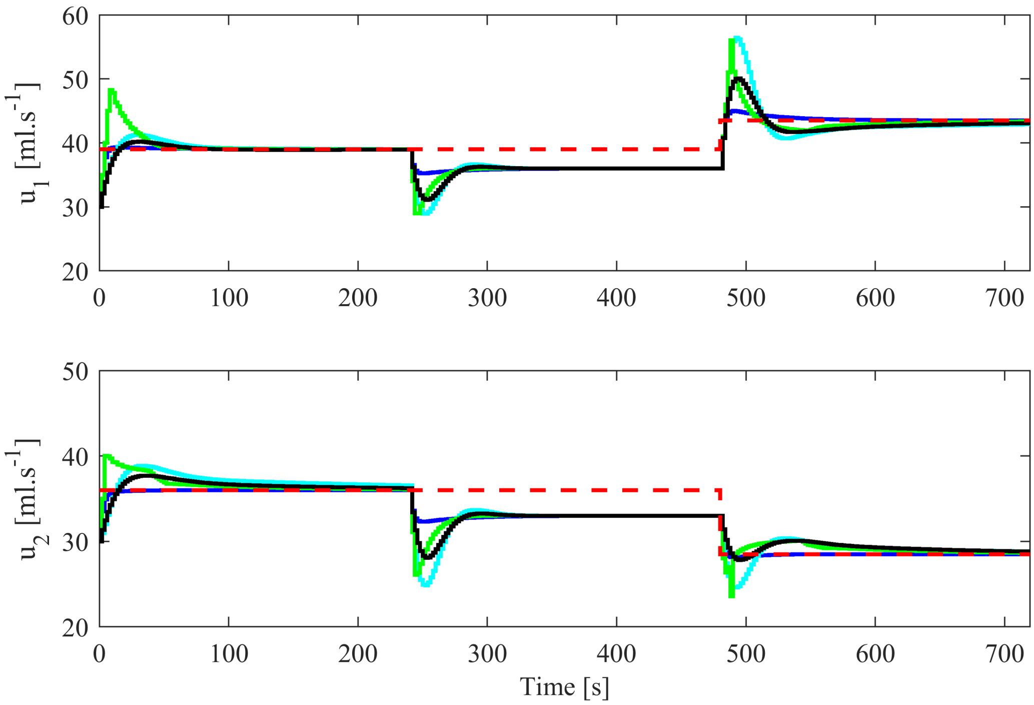


Source: Own author.

The controllers perform adequately in closed-loop regarding state trajectories. The first third of the simulation shows every controller bringing system states to the control zone, although the terminal contracting constraint controller shows overshoot from the start-up state, as it must contract the final stage cost w.r.t. the previous final stage cost. The second part shows similar behavior for every closed-loop state trajectory, but now the terminal equality constraint controller shows slight overshoot when compared to others. The third part of the simulation shows different behaviors for the first and last two states. The first two states show distinct overshoot for every formulation while the last two states show two distinct trends: quickly reaching steady states - as is done by the terminal inequality constraint controller - or a slow convergence to steady state, shown by the no guaranteed stability controller and terminal contracting constraint controller. For these last two states, the terminal equality constraint controller shows an intermediary behavior by abiding by the former then showing the latter behavior. Although these results show some of the closed-loop results of the formulations, further scrutiny by other means is necessary. Later in this work a sensitivity analysis regarding controller tuning is performed for every formulation deployed.

The computed closed-loop control policies are in Figure 52.

Figure 52 – Quadruple-tank system computed inputs for different controllers: NC (—), TCC (—), TEC (—), and TIC (—) formulations and input targets (- - -)



Source: Own author.

The formulations show similar behavior input-wise for all three parts of the simulation: no guaranteed stability formulation and terminal inequality constrained show the least overshoot, while terminal inequality constrained formulation is the quickest to converge its input. Terminal contracting constrained formulation shows the most overshoot, followed by the terminal equality constrained formulation.

New metrics are introduced in order to evaluate closed-loop performance by means other than visualizing closed-loop states or computed inputs. Since the formulations contain varied forms of objective functions, they cannot be compared via controller cost.

A proposed comparison is done by comparing the capability of bringing and keeping system states inside their control zones with the distance-to-zone performance index $D(\boldsymbol{x}, \mathscr{X}_z)$:

$$D(\boldsymbol{x}, \mathscr{X}_z) = \sqrt{d(\boldsymbol{x}, \mathscr{X}_z)^T \boldsymbol{I}_{n_x} d(\boldsymbol{x}, \mathscr{X}_z)} \tag{5.2}$$

$$d(\boldsymbol{x}, \mathscr{X}_z) = \begin{cases} 0, \text{ for } \boldsymbol{x} \in \mathscr{X}_z \\ \min(|\boldsymbol{x}_i - \mathscr{X}^i_{z,min}|, |\boldsymbol{x}_i - \mathscr{X}^i_{z,max}|), \text{ for } \boldsymbol{x} \notin \mathscr{X}_z, \end{cases} \tag{5.3}$$

which is zero when the state is in the control zone or the shortest distance to the control zone coordinate-wise, as indicated by expression (5.3). The weighted norm of this distance is then taken, as described by (5.2).

The closed-loop performance of different formulations is done as well by comparing their exerted control effort via the performance index $||\boldsymbol{\Delta u}||$:

$$||\boldsymbol{\Delta u}|| = \sqrt{\boldsymbol{\Delta u}^T \boldsymbol{I}_{n_u} \boldsymbol{\Delta u}} \tag{5.4}$$

Basically, the control effort is defined by the norm of the control moves taken by the controller, as depicted in (5.4). It is a metric to compare different control effort that different formulations show.

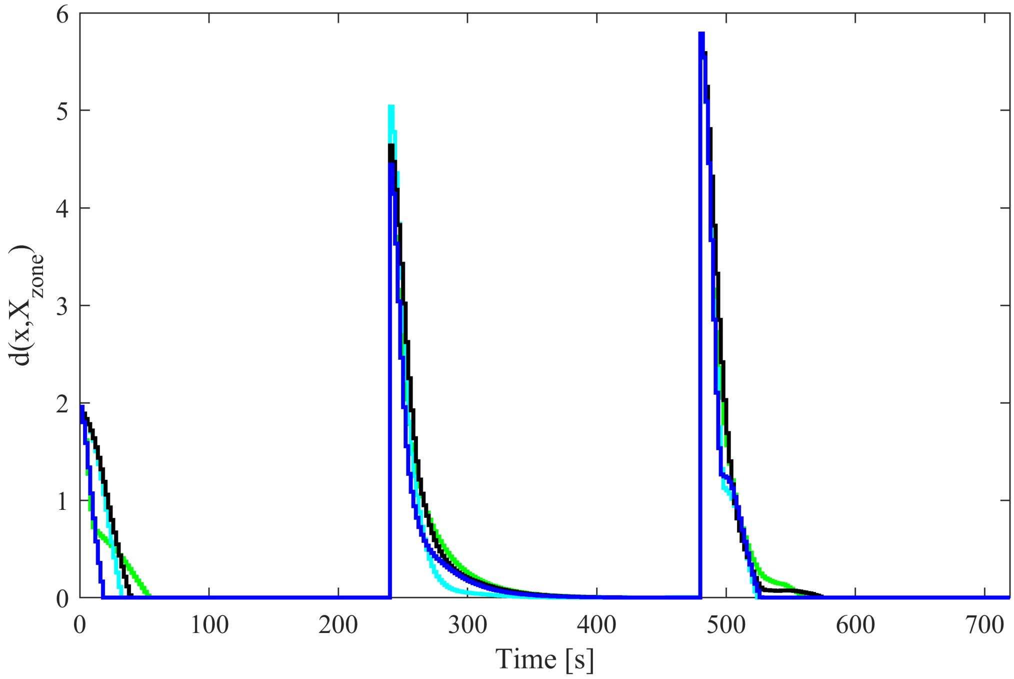
Finally, another comparison is drawn by measuring the deviation between computed inputs and the input target. The economic performance index:

$$||\boldsymbol{u} - \boldsymbol{u}_{des}|| = \sqrt{(\boldsymbol{u} - \boldsymbol{u}_{des})^T \boldsymbol{I}_{n_u} (\boldsymbol{u} - \boldsymbol{u}_{des})} \tag{5.5}$$

which is the norm of the distance between input $\boldsymbol{u}$ and economic target $\boldsymbol{u}_{des}$, intends on measuring how far the closed-loop is from the desired economic target.

First, distance-to-zone performance index for the multiple formulations deployed can be seen in Figure 53.

Figure 53 – Closed-loop distance-to-zone performance for different controllers: NC (—), TCC (—), TEC (—), and TIC (—) formulations



Source: Own author.

No guaranteed stability formulation shows the slowest dynamics of states towards the control zones in every condition, with the exception of start-up condition, being the last to bring the distance to zero. This formulation is heavily dependant on approaching an optimal control via extension of optimization horizon. The proposed 36 seconds of optimization horizon are smaller than the 54 seconds the terminal equality constraint formulation requires to be feasible, and lower than the reported 60 seconds required in order to avoid state drifting, as reported by Raff et al. (2006). Hence, due to its short optimization horizon, trajectories computed by this formulation are suboptimal trajectories and have no compensation for their suboptimality, unlike terminal inequality constraints that have terminal penalty as well the constraint itself. We have the sum of this distance over the entire simulation for each simulation in Table 2.

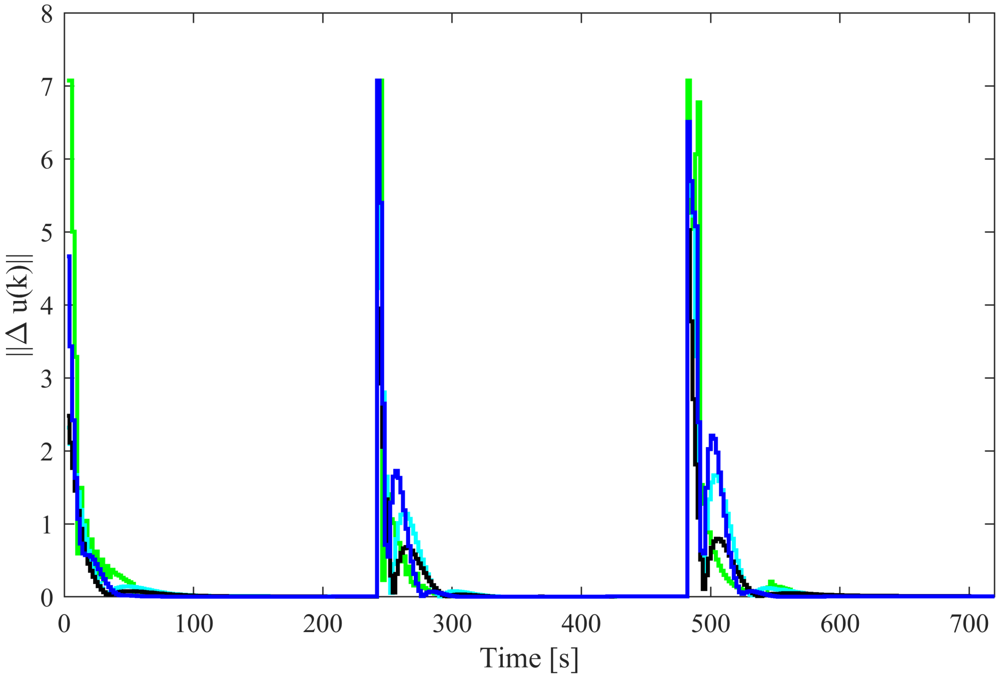
Table 2 – Zone tracking performance for different controller formulations

Terminal constraint	$\sum D(\boldsymbol{x}, \mathscr{X}_Z)$
Equality (TEC)	98.5
Inequality (TIC)	87.1
Contracting (TCC)	109.2
None (NC)	118.8

Source: Own author.

These sums show what is seen in Figure 53, that is, the no guaranteed stability formulation performs the worst when comparing state distance to control zone. As the terminal contracting constraint must be satisfied, speed of state towards control zone suffers, yielding the worse performance of all the constrained formulations. The terminal inequality constrained formulation performs the best in this metric, followed by the terminal equality constrained formulation. Next, control effort performance in Figure 54.

Figure 54 – Closed-loop control effort performance for different controllers: NC (—), TCC (—), TEC (—), and TIC (—) formulations



Source: Own author.

Regarding control effort, terminal contracting constrained formulation shows the highest control effort out of all of the formulations at the start of every zone or economic target change, which may be effect of terminal contracting constraint. The no guaranteed stability formulation shows the lowest control effort since there are no constraints to satisfy and due to its suboptimality, its control effort would not match an optimal control policy. Terminal equality constrained formulation would be one to have large control effort, but that is not the case. Since equilibrium must be achieved at the end of every optimization horizon, this formulation shows conservative control policies when compared to controllers with other constraints. With an additional comparison, Table 3 shows the sum of the control move norms for the different controller formulations.

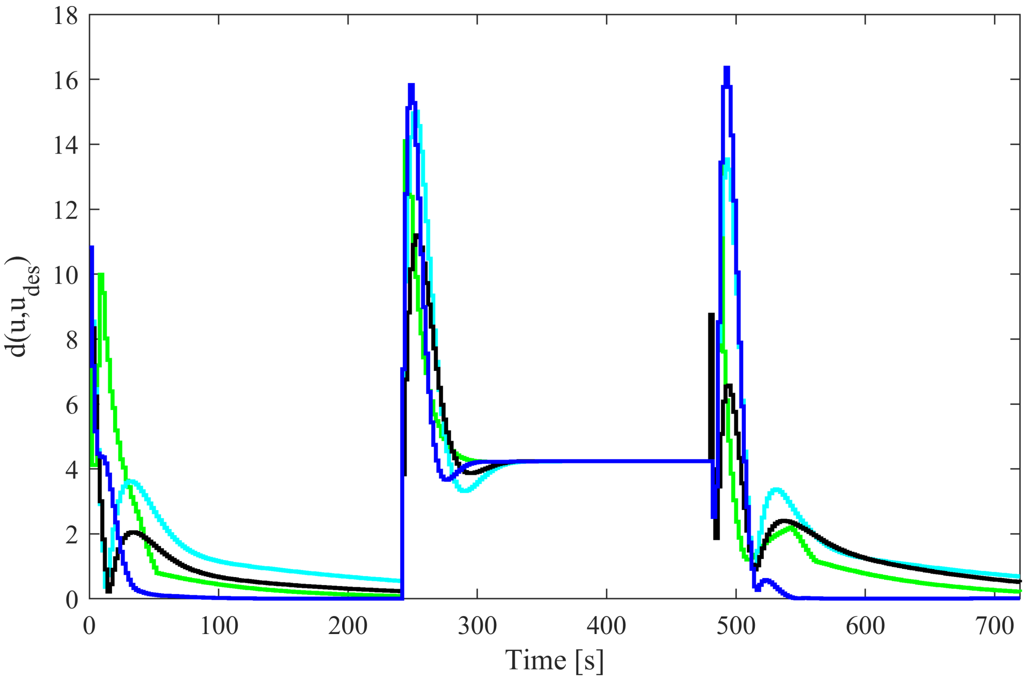
Table 3 – Actuator preservation performance for different formulations

Terminal constraint	$\sum \|\Delta \boldsymbol{u}\|$
Equality (TEC)	87.7
Inequality (TIC)	90.7
Contracting (TCC)	99.2
None (NC)	59.6

Source: Own author.

Sum of control move norm reinforces what was visualized in Figure 54. Out of all the constrained formulations, terminal equality constraint formulation is more conservative regarding control effort, while the terminal contracting constraint exerts the most control effort out of all formulations. Regarding all formulations, the one with the best performance in this regard is the no guaranteed stability formulation. Now, onto economic performance of these formulations, in Figure 55.

Figure 55 – Economic performance for different controllers: NC (—), TCC (—), TEC (—), and TIC (—) formulations



Source: Own author.

On economic performance, both terminal equality constrained and no guaranteed stability formulations show the slowest progress towards the economic target (when attainable), posing an interesting question regarding terminal equality constraints hindering economic performance when deployed along with control zones. The no guaranteed stability formulation's behavior could be explained by how sub-optimal trajectories yield poor performance, but for the terminal equality constraint formulation behavior, poor economic performance seems to depend on how distant the final closed-loop equilibrium point is from the control zone boundary. Terminal equality constrained formulation behaves differently when the economic target is not compatible with the control zone, showing considerable overshoot but converging faster to the economic target when compared to its convergence when the economic target is compatible with the zone. As the constraint must be satisfied at every sampling interval and reaching the reference value is a higher priority than achieving the economic target - which is only a penalty, its economic performance is affected negatively. Table 4 shows the sum of the distance-to-target performance index.

Table 4 – Sum of distance-to-target for different controller formulations

Terminal constraint	$\sum \|\boldsymbol{u} - \boldsymbol{u}_{des}\|$
Equality (TEC)	1064.10
Inequality (TIC)	788.52
Contracting (TCC)	880.92
None (NC)	896.20

Source: Own author.

Terminal contracting constrained formulation shows intermediary economic performance out of all of the implementations, close to the no guaranteed stability formulation. Terminal inequality constrained formulation, after the proposed adaptations, shows the best economic performance out of all the formulations and the terminal equality constrained formulation has the worst economic performance.

Another comparison between formulations is done, regarding computational effort each formulation demands. The closed-loop simulations were performed three times, with collection of used computer time by the optimizations. Then, the average time spent per iteration over the three simulations was computed. Finally, the average time spent per iteration over the entire simulation was computed from the average time spent per each iteration just computed. The average computer time spent per iteration can be seen in Table 5.

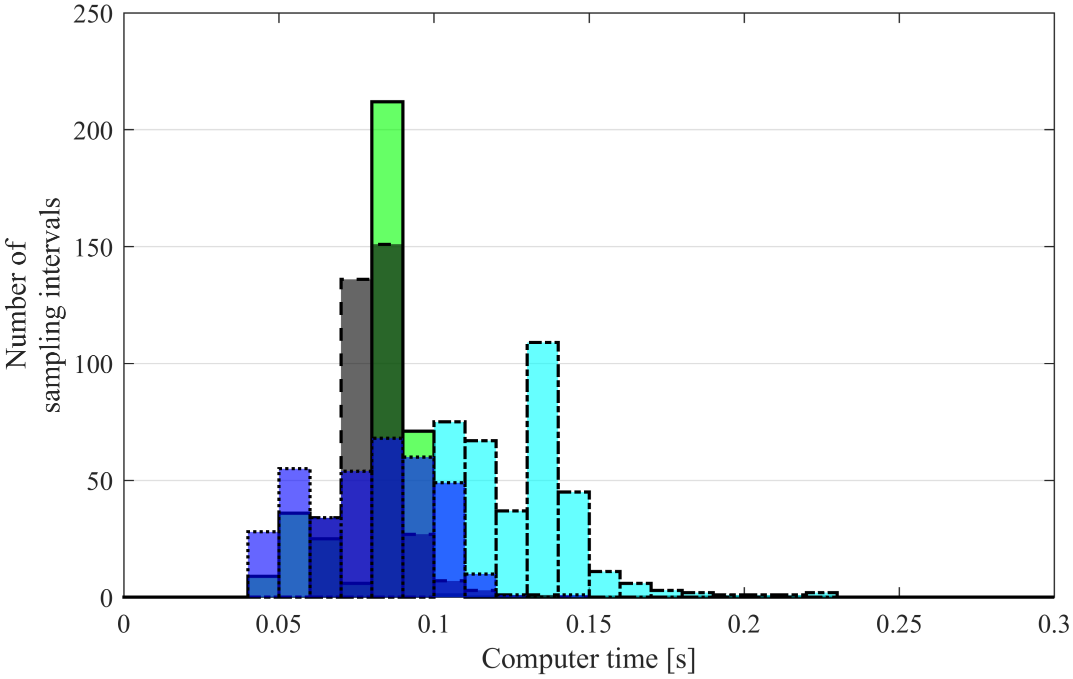
Table 5 – Average computer time spent per iteration over a simulation with standard deviation for different controller formulations

Terminal constraint	$\bar{t} \pm \varsigma$
Equality (TEC)	0.1289 ± 0.0190
Inequality (TIC)	0.0796 ± 0.0199
Contracting (TCC)	0.0810 ± 0.0130
None (NC)	0.0820 ± 0.0084

Source: Own author.

As expected, the terminal equality constraint formulation was more computationally demanding, with the largest mean time. All other formulations, with or without stabilizing constraints, show similar computer times. The mean computer time histogram for these controller formulations mean computer times, classified in intervals of 0.01s up to 0.3 seconds, can be seen in Figure 56.

Figure 56 – Mean computer time spent per iteration over three simulations histogram for different controllers: NC (■), TCC (🟩), TEC (▨), and TIC (🟦) formulations



Source: Own author.

Terminal equality constraint shows more violations of 10% of the sampling time threshold. It is followed by the terminal contracting constraint, with one violation at the first sampling interval. The equality constrained formulation's computational demand becomes more evident as its computer times distribution is almost entirely separated from other formulations. For applicability of this terminal equality constrained controller, this formulation was implemented with multiple shooting with numerical integration, and multiple shooting with orthogonal collocation over finite elements. The method of computer time collection and subsequent mean computer time calculation was the same: three simulations per different optimization implementation. Mean computer times with standard deviation can be seen in Table 6.

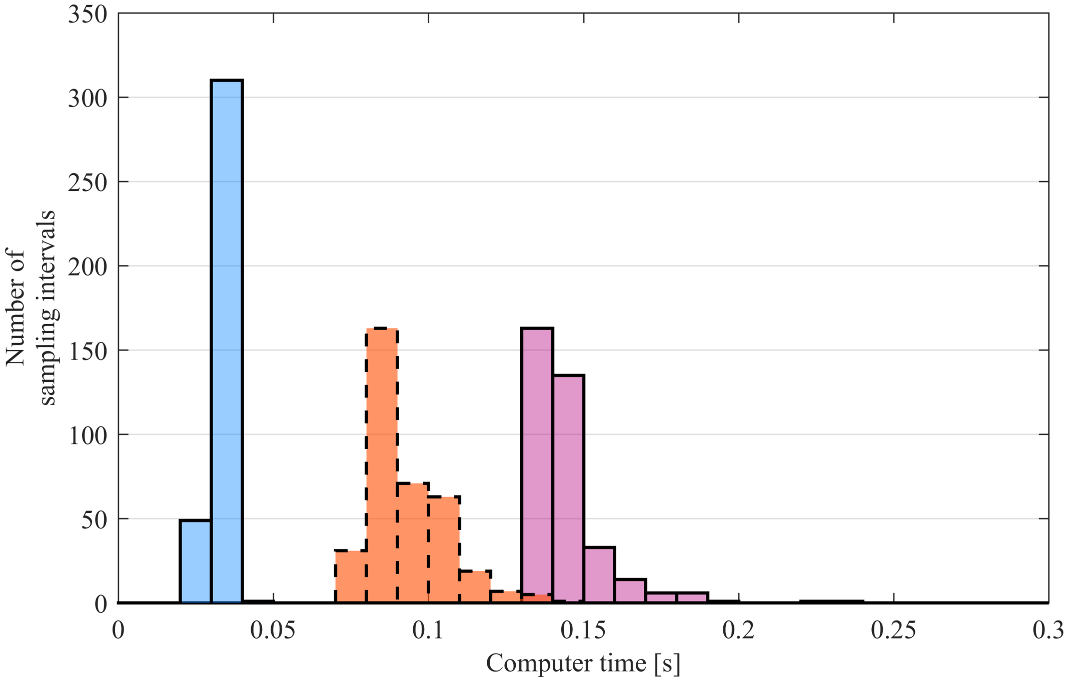
Table 6 – Mean computer times with standard deviation of TEC formulation for different shooting methods

Approach	$\bar{t} \pm \varsigma$
Single shooting	0.1440 ± 0.0124
Multiple shooting	0.0926 ± 0.0133
Orthogonal collocation	0.0324 ± 0.0024

Source: Own author.

Without change in shooting method, this formulation would be impractical (with its computer time threshold violations). With multiple shooting, there is a significant reduction in mean computer times as well as variance, matching all other formulations implemented with single shooting method. Further reduction was observed as the numerical integrator was substituted by a quadrature method. The discrepancies between these cases is evident in Figure 57.

Figure 57 – Overall performance for single shooting (🟪), multiple shooting (🟧) and orthogonal collocation (🟦)



Source: Own author.

Multiple shooting, in this situation, already shows satisfying results regarding computer times, with the startup as the only violation of the computer time threshold previously established. Quadrature method shows consistent, and even lower computer times. It is evident how almost third of the sampling times are higher due to how the control zones and input targets were selected: in the second part, the control zone is incompatible with the economic target.

5.2.1 Controller selection

This section will present the methodology developed for controller selection. It is based on the performance indexes exposed earlier. This work will then compare the formulations based on their zone tracking performance, actuator preservation and economic performance based on the distance-to-zone, control effort and distance-to-target indexes.

In order to compare the formulations, they were simulated in a different setting of the quadruple-tank system. The starting state was changed to $\boldsymbol{x}_0 = \begin{bmatrix} 20 & 20 & 20 & 20 \end{bmatrix}^T$ and the length of this simulation was shortened to $120t_{sampling}$. The zone and input targets still happen in the same manner as the one in the previous subsection.

The penalty matrices and/or prediction and control horizons of each controller was varied in different manners for each formulation.

The TEC formulation have had its state penalty matrix $\boldsymbol{Q}_x$ multiplied by 1, 5 and 10, input move penalty matrix $\boldsymbol{R}$ multiplied by 1, 3 and 9, and input target penalty matrix $\boldsymbol{Q}_u$ multiplied by 1, 2 and 3, in order to show cases where zone tracking, control effort or input target tracking were prioritized by the practitioner. Control and prediction horizon lengths of 30, 40, and 50 were also considered.

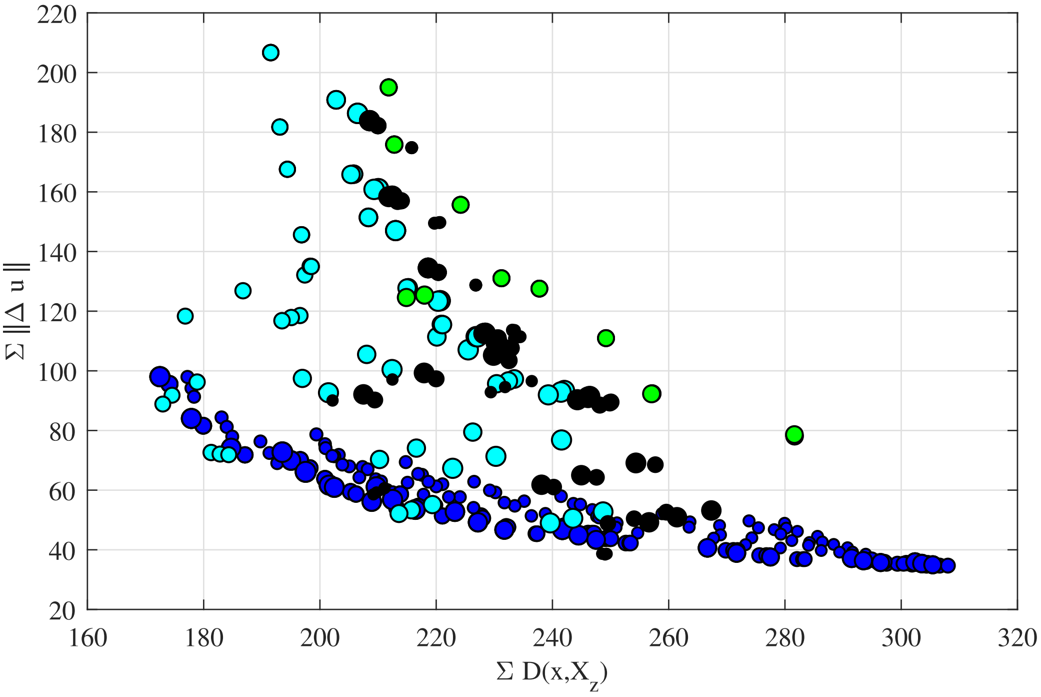
The TIC formulation have had its state penalty matrix $\boldsymbol{Q}_x$ multiplied by 1, 5 and 10, input move penalty matrix $\boldsymbol{R}$ multiplied by 1, 4 and 9, and input target penalty matrix $\boldsymbol{Q}_u$ multiplied by 1, 2 and 3, with equal control and prediction horizon lengths of 10, 15, and 20 sampling intervals. Its terminal region radius was 0.8, 0.9, and 1.

As for the TCC formulation have had state penalty matrix $\boldsymbol{Q}_x$ multiplied by 1, 5 and 10, input move penalty matrix $\boldsymbol{R}$ multiplied by 1, 4 and 9, and input target penalty matrix $\boldsymbol{Q}_u$ multiplied by 1, 2 and 3. Its control horizon and prediction horizon were increased to 9 and 18, respectively, since there was a change in the starting state and the previous horizon lengths were infeasible.

Finally, the NC formulation have had horizon length N of 10, 20, and 30. State error penalty matrix $\boldsymbol{Q}_x$ was multiplied by 1, 5, and 10, input move penalty matrix $\boldsymbol{R}$ multiplied by 1, 4, and 9, and input target penalty matrix $\boldsymbol{Q}_u$ multiplied by 1, 2 and 3.

These formulations with varied tuning parameters were simulated five times and have had their computer time spent per iteration measured as well, with the mean value taken to account for run-to-run variations. Terminal inequality constrained formulation did not account for terminal region computation nor target calculation times. The results can be seen in Figures 58 to 61.

Figure 58 – Distance-to-zone and control effort of different controllers: NC (●), TCC (●), TEC (●), and TIC (●) formulations. Marker size scales with computer times.



Source: Own author.

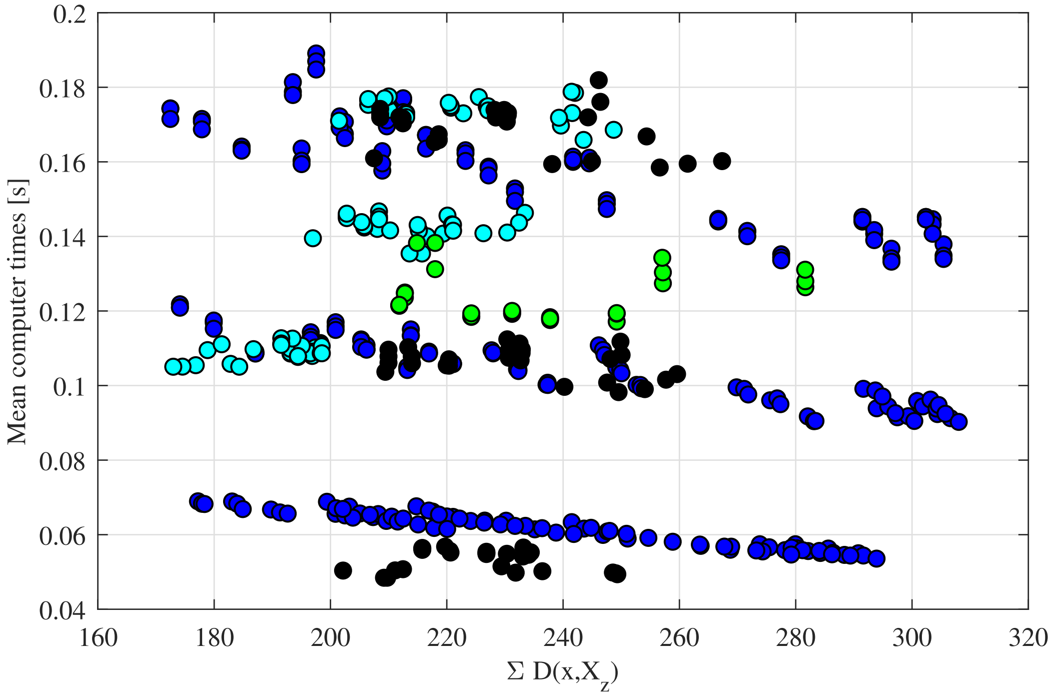
The TIC formulation is capable of a wide range of performance regarding zone tracking, as can be seen with the spread of distance-to-zone. The NC formulation and the TEC formulations show similar ranges of zone tracking performance, although the NC formulation cannot reach distance-to-zone values as low as the TEC formulation. The TCC formulation is not as versatile as other formulations in this regard, showing a very narrow range of distance-to-zone performance.

Regarding the control effort, the TCC formulation has shown to be not conservative with its actuators (indicated by its higher control effort values observed in the simulations), whereas the TIC formulation shows the least control effort values, with the narrowest range of possible control effort values out of all the formulations. Both the NC and the TEC formulation have shown its capacity of a wide range of control effort performance.

Regarding the trade-off between control effort and zone tracking performance, the TIC formulation shows a behavior similar to an asymptotic curve. The TEC formulation and the NC formulation show a spread of trade-off in this regard, warranting care from the practitioner as to consider that there might be considerable losses when considering only a single aspect of the controller.

As for computer times, all of the formulations show regions where the zone tracking and control effort performance is similar while there is a difference in computer times, which also indicates caution is advised when tuning all of the formulations exposed. This is best seen in Figure 59.

Figure 59 – Distance-to-zone and computer times of different controllers: NC (●), TCC (●), TEC (●), and TIC (●) formulations.

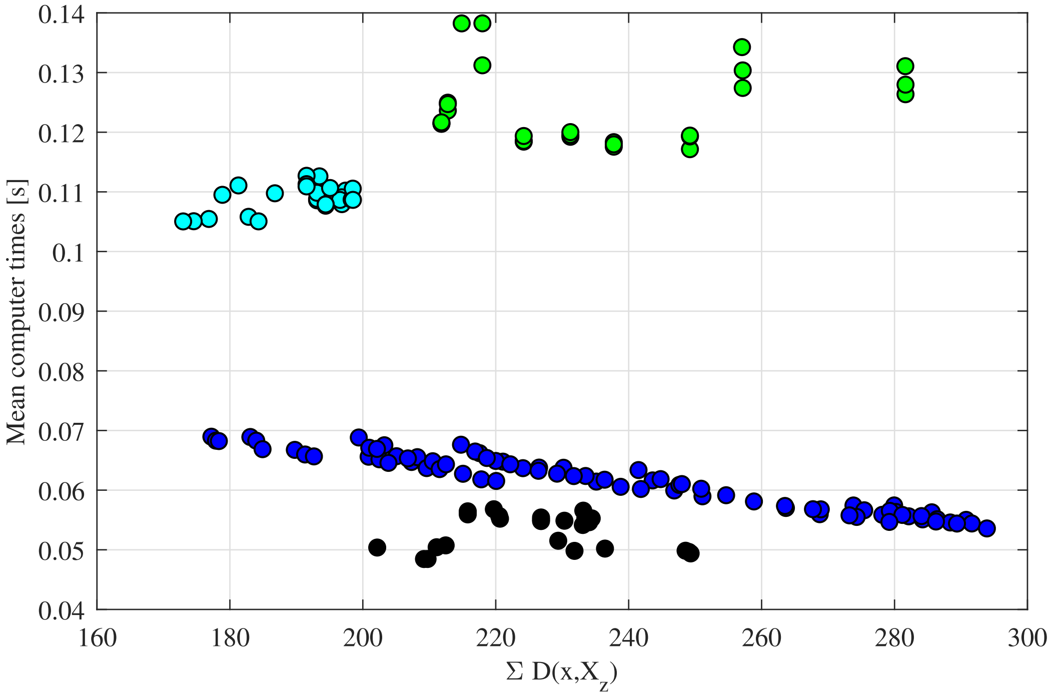


Source: Own author.

While the TCC formulation does not show this split behavior (due to the single horizon length chosen), the other formulations have its sets of data split into three parts. This is due to the multiple horizon lengths chosen. The TIC formulation shows this behavior regardless of magnitude of terminal ellipsoid radius given to the controller. The NC formulation is capable of delivering the fastest zone tracking control with average performance. The terminal equality constraint improves zone tracking performance at the expense of computer times, yielding similar performance to the TIC formulation with higher computational effort. Increasing horizons might detract from zone tracking performance for the TIC formulation, TEC formulation and NC formulation, with a significant increase in computational effort. If one were to consider only the simulations with the shortest horizon lengths for all formulations, these would be ordered simply by their computational demands. This is seen in Figure 60.

Considering the previously observed results for the TCC formulation, it is clear that this formulation is sensitive to start-up. The change in starting state has required increased control and prediction horizons, yielding larger computer times than observed at the previous simulation. The TEC formulation has been implemented with a multiple-shooting method, in order to lower its computer times - which could be even lower if the practitioner would consider orthogonal collocation over finite elements instead of a numerical integrator in the state prediction step of the optimization problem. TIC and NC formulations show the shortest computer times with similar performance, although the former seems to be more flexible regarding multiple objectives (due to its range of zone tracking performance). This is further investigated by comparing the

Figure 60 – Distance-to-zone and computer times of different controllers: NC (●), TCC (🟢), TEC (○), and TIC (🔵) formulations. Shortest horizon lengths only.



Source: Own author.

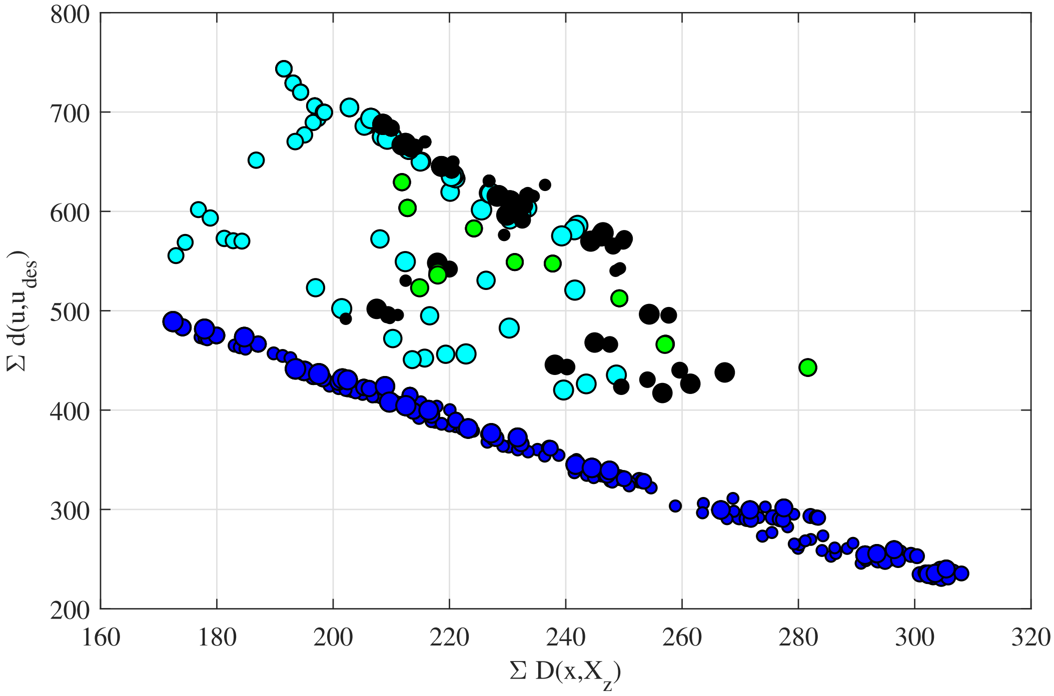
zone tracking performance with the input target tracking performance in Figure 61.

Considering economic performance alone, every formulation has shown some flexibility in its multi-objective aspect, given how they are able to prioritize zone tracking performance or economic performance when proper weights are applied. However, the TIC formulation is capable of delivering better economic performance, considering it is the only capable of obtaining economic performance indexes under the range of 400.

For the trade-off aspect, every formulation shows a trend of declining economic performance as zone tracking performance improves. The formulation which shows this behavior the best is the TIC formulation. The worst possible economic performance is delivered by the TEC formulation, closely followed by the NC formulation. These results put into question the necessity of optimizing targets as one of the objectives the controllers must follow.

The TIC formulation is the only one completely incapable of accepting an input target, given the adaptation shown in this work that consists of an additional target calculation layer which delivers a set point that belongs to the current control zone and minimizes the distance between steady-state input value and the desired input target, as per Problem $\mathscr{P}_d$. Since this formulation has a fixed set point - computed at every control zone change and/or input target change - the overall economic performance of all formulations point to the possibility that the set point as a decision variable might not be economically beneficial in general.

Figure 61 – Distance-to-zone and input target tracking of different controllers: NC (●), TCC (🟢), TEC (○), and TIC (🔵) formulations. Marker size scales with computer times.



Source: Own author.

6 Conclusion

This work concluding remarks will go over the approached topics and their demonstrated contribution to the proposed theme.

Knowledge about the system being worked on is as important for a nonlinear mathematical model as it is for a linear mathematical model. Even more so as nonlinear models cannot have infinite horizon predictions computed easily, or have the same behavior over the entire state space. Usage of functional analysis is mostly limited to identifying candidate Lyapunov functions for controller cost or evaluating eigenvalues point-wise, leading to rather poor understanding of the studied system. It has been shown in this work the conjoint use of Lie algebra and parametric continuation in determining equilibrium points. The NMPC formulations implemented in this work rely on some overlap between the limit set and the control zone, so enabling determination of equilibria in situations when open-loop simulations are incapable and enables the use of NMPCs in systems with unstable regions without violating any conditions for closed-loop stability.

Better understanding of invariant sets was required for NMPC formulations associated with terminal regions, so a through exploration of feedback stabilization via LQR was also done, considering the discrete-time case. Methods from the literature were extended to the case where input moves are bounded. The trade-off between using SDP and LMI methods was shown. The former has better closed-loop performance due to its larger gains, while the latter yields larger operating legions - an interesting trade considering cases where one focuses on closed-loop performance or applicability in cases where the regulator itself is virtual.

Regarding NMPC formulations, their characteristics were explored regarding their computer times and a preview of their performance by visual comparison of state and input profiles over time. Regarding their computational effort, both the terminal inequality constrained and the terminal contracting constrained formulations show growing computational requirements as the states approach reference value. This behavior can be explained by how both optimization problems approach a terminal equality constrained formulation as the state trajectories converge to the reference. Meanwhile, the terminal equality constrained formulation has the largest overall computer times - with possible competitiveness in this regard if one were to further alter shooting methods and integration methods, and the no guaranteed stability formulation has the smallest computational demand.

The multiple formulations used in this work are closed-loop stabilizing and recursively feasible, they can have their closed-loop performance compared not directly by controller cost, but through the proposed metrics. Different constraints lead to varying performance regarding different metrics. Terminal contracting constraints were less conservative when it comes to control effort, while terminal equality constraints were more conservative in this regard. The formulation without guaranteed stability was the worst when it comes to bringing state to the control zone, related to how it exerts the least control effort. The best formulation to bring states to control zones was between the terminal inequality constrained formulation (for references far from the control zone frontier) and the terminal equality constrained one (for references near the control zone frontier). Finally, economic performance was intermediary with the terminal contracting constraint formulation, with the best economic performance coming from terminal inequality constraint formulation and the worst varying between the no guaranteed stability and the terminal equality constrained formulations. The implementation of terminal equality constrained formulations with economic term is questionable, given the poor economic performance of the terminal equality constrained formulation shown. The dominance of the terminal inequality constrained formulation, the only two-layer formulation in the comparison, puts into question the necessity of one-layer controllers with the intention of improving economic performance.

With the effects of different formulations on different performance indexes in mind, a judicious choice of formulation can lead to less complex solutions with less computational effort and acceptable performance. Regarding computational effort, switching shooting method has shown considerable computational effort reductions, with further reductions accompanying change of a numerical integrator into a quadrature method. Advancements in computational power and optimization algorithms will eventually lead to IHOCPs being solved online. As this is not the current situation, discussions of near-optimal, feasible and stabilizing formulations and their applicability are still pertinent.

There is room for more studies regarding differential geometry's applications in nonlinear system analysis and, by consequence, NMPC. Assessment of how necessary some concepts of functional analysis are to NMPC, such as dual vector spaces, as these concepts are already integrated with the field of optimization itself.

The formulations presented in this work should be able to function along with nonlinear models generated through means other than first-principle. Mathematical models generated through artificial intelligence, as an example, should be able to work as a state prediction method while these models represent the plant properly as well as satisfy some assumption required by the formulation itself - for example, a continuously differentiable nonlinear model or the existence of a stabilizable Jacobian linearization of the nonlinear model.

This work handled nominal controllers only. Although some of the formulations' constraints were designed in order to be not as restrictive as an equality constraint, no cases with persistent disturbance - be it noise or model mismatch - were considered. As future work, comparisons between robust controllers can be done.

Bibliography

AGRACHEV, A. A.; SACHKOV, Y. L. *Control Theory from the Geometric Viewpoint*. Berlin, Heidelberg: Springer, 2004. Cited in page 143.

ALAMIR, M. Stability proof for nonlinear MPC design using monotonically increasing weighting profiles without terminal constraints. *CoRR*, 2017. Cited in page 104.

ANDERSSON, J. A. E. et al. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, Springer, v. 11, n. 1, p. 1–36, 2019. Cited in page 31.

CANNON, M.; DESHMUKH, V.; KOUVARITAKIS, B. Nonlinear model predictive control with polytopic invariant sets. *Automatica*, v. 39, n. 8, p. 1487–1494, 2003. ISSN 0005-1098. Cited in page 88.

CHEN, H.; ALLGÖWER, F. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, v. 34, n. 10, p. 1205 – 1217, 1998. Cited 4 times in page 49, 88, 94, and 99.

DHOOGE, A. et al. New features of the software MatCont for bifurcation analysis of dynamical systems. *Mathematical and Computer Modelling of Dynamical Systems*, 2008. Cited 2 times in page 35 and 41.

DUAN, G.; YU, H. LMIs in Control Systems: Analysis, Design and Applications. In: . [S.l.: s.n.], 2013. Cited in page 54.

ECONOMOU, C. G.; MORARI, M.; PALSSON, B. O. Internal model control: extension to nonlinear system. *Industrial & Engineering Chemistry Process Design and Development*, v. 25, n. 2, p. 403–411, 1986. Cited in page 29.

FAGIANO, L.; TEEL, A. Generalized terminal state constraint for model predictive control. *Automatica*, v. 49, 07 2012. Cited in page 87.

GARCIA, C. E.; MORSHEDI, A. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, Taylor & Francis, v. 46, n. 1-3, p. 73–87, 1986. Cited in page 29.

GIRALDO, S. A.; MELO, P. A.; SECCHI, A. R. Tuning of model predictive control based on hybrid optimization. *IFAC-PapersOnLine*, v. 52, n. 1, p. 136 – 141, 2019. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019. Cited in page 84.

GONZÁLEZ, A. H.; ODLOAK, D. A stable MPC with zone control. *Journal of Process Control*, v. 19, n. 1, p. 110 – 122, 2009. Cited 3 times in page 80, 94, and 109.

GRIFFITH, D. W.; BIEGLER, L. T.; PATWARDHAN, S. C. Robustly stable adaptive horizon nonlinear model predictive control. *Journal of Process Control*, v. 70, p. 109 – 122, 2018. ISSN 0959-1524. Cited in page 98.

GRÜNE, L. NMPC without terminal constraints. *IFAC Proceedings Volumes*, v. 45, n. 17, p. 1 – 13, 2012. 4th IFAC Conference on Nonlinear Model Predictive Control. Cited in page 110.

GRÜNE, L.; PANNEK, J. *Nonlinear Model Predictive Control: Theory and Algorithms*. [S.l.]: Springer London, 2011. (Communications and Control Engineering). ISBN 9780857295019. Cited in page 110.

HAHN, W. *Stability of Motion*. [S.l.]: Springer Berlin Heidelberg, 1967. Cited in page 73.

HINDMARSH, A. C. et al. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, ACM, v. 31, n. 3, p. 363–396, 2005. Cited in page 31.

KEERTHI, S.; GILBERT, E. G. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, Springer, v. 57, n. 2, p. 265–293, 1988. Cited 2 times in page 76 and 77.

KOTHARE, S. L. de O.; MORARI, M. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, v. 45, n. 6, p. 1053–1071, 2000. Cited in page 99.

KUMAR, M. P.; KAISTHA, N. Role of multiplicity in reactive distillation control system design. *Journal of Process Control*, Elsevier, v. 18, n. 7-8, p. 692–706, 2008. Cited in page 42.

LÖFBERG, J. YALMIP: A toolbox for modeling and optimization in matlab. In: *In Proceedings of the CACSD Conference*. Taipei, Taiwan: [s.n.], 2004. Cited 2 times in page 31 and 60.

MEJÍA, J. S.; STIPANOVIĆ, D. M. A modified contractive model predictive control approach. In: *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. [S.l.: s.n.], 2009. p. 1968–1973. Cited in page 104.

MICHALSKA, H.; MAYNE, D. Q. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, IEEE, v. 38, n. 11, p. 1623–1633, 1993. Cited in page 88.

MORSHEDI, A. M.; CUTLER, C. R.; SKROVANEK, T. A. Optimal solution of dynamic matrix control with linear programing techniques (ldmc). In: *1985 American Control Conference*. [S.l.: s.n.], 1985. p. 199–208. Cited in page 29.

MUSKE, K. R.; RAWLINGS, J. B. Model predictive control with linear models. *AIChE Journal*, v. 39, n. 2, p. 262–287, 1993. Cited in page 75.

OLVER, P. J. *Applications of Lie Groups to Differential Equations*. New York: Springer-Verlag, 1993. Cited 2 times in page 143 and 147.

PARK, S. W. *Aplicação de controladores geométricos não-lineares em processos químicos*. PhD Thesis (PhD Thesis) — Universidade de São Paulo, Escola Politécnica, Curso de Pós-Graduação em Engenharia Química, São Paulo, 1995. Cited in page 142.

RAFF, T. et al. Nonlinear model predictive control of a four tank system: An experimental stability study. In: IEEE. *2006 IEEE International Conference on Control Applications*. [S.l.], 2006. p. 237–242. Cited 3 times in page 59, 65, and 121.

RAJHANS, C. et al. Terminal region characterization and stability analysis of discrete time quasi-infinite horizon nonlinear model predictive control. *Journal of Process Control*, v. 83, p. 30–52, 11 2019. Cited 8 times in page 55, 58, 88, 89, 90, 93, 94, and 98.

RICE, R.; DO, D. *Applied Mathematics And Modeling For Chemical Engineers*. [S.l.]: Wiley, 2012. ISBN 9781118024720. Cited in page 107.

RUSSO, L. P.; BEQUETTE, B. W. Impact of process design on the multiplicity behavior of a jacketed exothermic CSTR. *AIChE Journal*, Wiley Online Library, v. 41, n. 1, p. 135–147, 1995. Cited 3 times in page 38, 39, and 42.

SCOKAERT, P.; RAWLINGS, J. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, v. 43, n. 8, p. 1163–1169, 1998. Cited in page 52.

SENCIO, R. R. et al. A terminal state contractive nonlinear mpc with output zones and input targets. *IFAC-PapersOnLine*, v. 53, n. 2, p. 6025–6030, 2020. ISSN 2405-8963. 21st IFAC World Congress. Cited 2 times in page 81 and 99.

SONTAG, E. D. *Mathematical Control Theory*. New York, NY: Springer, 1998. Cited in page 141.

SOUZA, G. A. S. d. et al. Static process operability analysis through Lie algebra. In: *23^o Congresso Brasileiro de Engenharia Química*. [S.l.: s.n.], 2021. Cited in page 38.

STURM, J. F. Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, v. 11, n. 1-4, p. 625–653, 1999. Available at: <https://doi.org/10.1080/10556789908805766>. Cited 2 times in page 31 and 60.

SUSSMANN, H. J.; JURDJEVIC, V. Controllability of nonlinear systems. *Journal of Differential Equations*, v. 12, n. 1, p. 95–116, 1972. ISSN 0022-0396. Cited in page 37.

SZNAIER, M.; DAMBORG, M. J. Suboptimal control of linear systems with state and control inequality constraints. In: *26th IEEE Conference on Decision and Control*. [S.l.: s.n.], 1987. v. 26, p. 761–762. Cited in page 52.

THE MATHWORKS, INC. *MATLAB version 9.3.0.713579 (R2017b)*. Natick, Massachusetts, 2017. Cited 4 times in page 31, 40, 41, and 60.

TÜTÜNCÜ, R.; TOH, K.; TODD, M. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, v. 95, n. 1, p. 189–217, 2003. Cited 2 times in page 31 and 60.

VIDYASAGAR, M. *Nonlinear Systems Analysis*. USA: Prentice-Hall, Inc., 1993. Cited in page 37.

WÄCHTER, A.; BIEGLER, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, Springer, v. 106, n. 1, p. 25–57, 2006. Cited in page 31.

XIE, F.; FIERRO, R. First-state contractive model predictive control of nonholonomic mobile robots. In: *2008 American Control Conference*. [S.l.: s.n.], 2008. Cited in page 99.

Appendix

APPENDIX A – Differential geometry

This appendix was written with intent of clarifying the origin of the expression (2.15), used to compute the limit set. The theoretical background is considerably different from the overall theoretical results presented in this work, so in order to preserve the flow of this work, the theoretical background involving differential geometry was moved to an appendix. In this appendix, there are many definitions from differential geometry which were used in order to arrive at the expression (2.15), such as manifolds (geometric object that represents the state space), Lie groups (geometric object which represents the evolution of the state in the manifold, which is a manifold in itself) and Lie algebras (geometric object which generates Lie groups, where the generators come from the dynamic system itself).

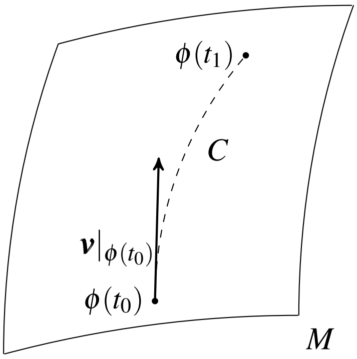
A.1 Manifolds

A convenient geometric object to perform nonlinear system studies is the manifold. In control systems literature, it is often replaced by a locally Euclidean space (SONTAG, 1998), and this substitution is often successful since very few analyses are done to account for the global behavior of a system. The advantage of working with a manifold comes from detaching the system of local coordinates $\boldsymbol{x} = (x^1, x^2, \ldots, x^{n_x})$ for a system in a space $\mathbb{R}^{n_x}$. Several structure characteristics of a manifold can be obtained. The manifolds used in this work are considered smooth (infinitely differentiable) and connected (cannot be represented by disjointed open sets).

A.2 Curves and tangent vectors

The evolution of a control system over time can be represented by curves. A curve C in a smooth manifold M is parametrized by a map $\phi : I \to M$, where I is a subinterval of $\mathbb{R}$. For dynamic systems, this interval usually represents a time interval. In local coordinates, this curve is represented by n_x functions $\boldsymbol{x} = \phi(t) = (\phi^1(t), \ldots, \phi^{n_x}(t))$. A curve is closed when its endpoints coincide: $\phi(t_1) = \phi(t_2)$, with the closed time interval $[t_1, t_2]$. This definition is important when it comes to define if a system is in equilibrium in a different manner as done by Definition 1. An illustration of these objects can be seen in Figure 62.

Figure 62 – Smooth manifold M with a curve C and a vector $\boldsymbol{v}$ tangent to C at a point $\phi(t_0)$



Source: Own author.

As the definition of a dynamic system given in Equation (2.10) is rather generic, we will now consider control-affine systems such as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{i=1}^{n_u} g_i(\boldsymbol{x})u_i \tag{A.1}$$

Say the dynamics of the system (A.1) are represented by the curve C on a manifold M, parametrized by the map $\phi = (\phi^1(t), \ldots, \phi^{n_x}(t))$. We have that at each point $x \in M$ there is a tangent vector to C, given by the derivative $\dot{\phi} = (\dot{\phi}^1(t), \ldots, \dot{\phi}^{n_x}(t))$. In local coordinates, the tangent vector $\boldsymbol{v}|_x$ to $C = \boldsymbol{\phi}(t)$ at point $x \in M$ would be

$$\boldsymbol{v}|_x = \dot{\boldsymbol{\phi}}(t) = \dot{\phi}^1(t)\frac{\partial}{\partial x^1} + \cdots + \dot{\phi}^{n_x}(t)\frac{\partial}{\partial x^{n_x}}. \tag{A.2}$$

Note that usually, vector fields are not represented with their respective differentials, i.e., in vertical form and ommiting the differential

$$\boldsymbol{v}|_{\boldsymbol{x}} = \dot{\boldsymbol{\phi}}(t) = [\dot{\phi}^1(t),\ \ldots,\ \dot{\phi}^{n_x}(t)]^T. \tag{A.3}$$

This is a common notation abuse in local coordinates, as pointed out in Park (1995), and may conflict with the concept of Lie bracket, or commutator, to be exposed later.

The collection of all tangent vectors to all possible curves that go through a point $\boldsymbol{x} \in M$ is called tangent space, $TM|_{\boldsymbol{x}}$, with same dimension as the manifold it is tangent to. The set of all tangent spaces to all points $\boldsymbol{x} \in M$ is called a tangent bundle. Finally, a vector field $\boldsymbol{v}$ in M determines a tangent vector $\boldsymbol{v}|_{\boldsymbol{x}}$ at every $\boldsymbol{x} \in M$.

A.3 Integral curves and flows

An integral curve ϕ of a vector field $\boldsymbol{v}$ is a smooth curve parametrized by t which has tangent vectors that coincide with the tangent vectors $\boldsymbol{v}|_{\boldsymbol{x}}$ determined by the vector field at the same point, that is,

$$\dot{\boldsymbol{\phi}}(t) = \boldsymbol{v}_{\boldsymbol{\phi}(t)}. \tag{A.4}$$

In local coordinates, $\boldsymbol{x} = \phi(t) = (\phi^1(t), \ldots, \phi^{nx}(t))$ is a solution to the ordinary differential equation (ODE) system

$$\frac{dx^i}{dt} = v^i(\boldsymbol{x}),\ i = 1, \ldots, nx, \tag{A.5}$$

where $v^i(\boldsymbol{x})$ represents the i-th component of the vector field $\boldsymbol{v}|_{\boldsymbol{x}}$.

The functions $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}_i(\boldsymbol{x})$, $i = 1, \ldots, n_u$, will be referred to as vector fields in the same manner. These vector fields determine an integral curve of the dynamic system, parametrized by time t, that represents the dynamic system states over time.

The existence of an integral curve implies that there is a maximal integral curve, which contains all other integral curves. This maximal integral curve is referred to as a flow, $\psi : \mathbb{R} \times \mathbb{R}^{nx} \to \mathbb{R}^{nx}$. The flow has the following properties:

$$\psi(t_2, \psi(t_1, \boldsymbol{x})) = \psi(t_1 + t_2, \boldsymbol{x}), \tag{A.6}$$

$$\psi(0, \boldsymbol{x}) = \boldsymbol{x}, \tag{A.7}$$

and

$$\frac{d}{dt}\psi(t, \boldsymbol{x}) = \boldsymbol{v}|_{\psi(t,\boldsymbol{x})} \tag{A.8}$$

for all t where defined.

A.4 Lie groups and flows

Lie theory is necessary since it allows the same approach that is done in linear system theory regarding controllability and observability, by transforming a nonlinear system into a linear system locally. With concepts such as groups and algebras, Lie theory has been used across many fields of study and has been groundwork for many other lines of work such as differential equation solving (OLVER, 1993) and classical mechanics (AGRACHEV; SACHKOV, 2004).

Although foreign to process systems engineering, the concept of groups are the base for theoretical work in Lie group theory. A group G can be defined as a set of elements joined with a group multiplication $m(\cdot, \cdot)$, such that for any two of its elements the result of the group multiplication is another element of the group G. The multiplication must satisfy the axioms of associativity:

$$m(n,m(o,p)) = m(m(n,o),p), \quad n,o,p \in G, \tag{A.9}$$

identity element (which will be referred to as Id):

$$m(Id,n) = n = m(n,Id), \quad n,Id \in G, \tag{A.10}$$

and inverses (and when there are no inverses, the set of elements is named a semi-group):

$$m(n,n^{-1}) = Id = m(n^{-1},n), \quad n,n^{-1},Id \in G, \tag{A.11}$$

One approachable example of group is the set of real numbers $\mathbb{R}$ with addition as its group multiplication and the scalar 0 as its identity element.

The flow generated by a vector field is identical to the flow generated by a local action of a Lie group G of one parameter t in a manifold M. This group is also a manifold in itself, and is commonly referred to as one-parameter group of transformations while the vector fields are referred to as the infinitesimal generators of this group. These infinitesimal generators can be obtained by:

$$\boldsymbol{v}|_{\boldsymbol{x}} = \left.\frac{d}{dt}\right|_{t=0} \psi(t,\boldsymbol{x}) \tag{A.12}$$

The action of the one-parameter Lie group can be represented by the matrix exponential function. The resulting flow ψ of the vector field $\boldsymbol{v}$ is represented by

$$\psi(t,\boldsymbol{x}) \equiv \exp(t\boldsymbol{v})\boldsymbol{x}. \tag{A.13}$$

With this notation, the flow properties are rewritten.

$$\exp[(t_1+t_2)\boldsymbol{v}]\boldsymbol{x} = \exp(t_1\boldsymbol{v})\exp(t_2\boldsymbol{v})\boldsymbol{x} \tag{A.14}$$

when defined,

$$\exp(\boldsymbol{0}\boldsymbol{v})\boldsymbol{x} = \boldsymbol{x}, \tag{A.15}$$

and

$$\frac{d}{dt}[\exp(t\boldsymbol{v})\boldsymbol{x}] = \boldsymbol{v}|_{\exp(t\boldsymbol{v})\boldsymbol{x}} \tag{A.16}$$

for all $\boldsymbol{x} \in M$.

A.5 Lie bracket

The Lie group operation is known as Lie bracket or commutator. If $\boldsymbol{v}$ and $\boldsymbol{w}$ are vector fields in M, its Lie bracket $[\boldsymbol{v},\boldsymbol{w}]$ is the unique vector field which satisfies:

$$[\boldsymbol{v},\boldsymbol{w}](f) = \boldsymbol{v}[\boldsymbol{w}(f)] - \boldsymbol{w}[\boldsymbol{v}(f)] = \boldsymbol{v} \circ \boldsymbol{w}(\boldsymbol{f}) - \boldsymbol{w} \circ \boldsymbol{v}(\boldsymbol{f}) \tag{A.17}$$

for every smooth function $f : M \to \mathbb{R}$. The bracket has the properties of bilinearity:

$$[c\boldsymbol{v} + c'\boldsymbol{v}', \boldsymbol{w}] = c[\boldsymbol{v},\boldsymbol{w}] + c'[\boldsymbol{v}',\boldsymbol{w}] \tag{A.18}$$

$$[\boldsymbol{v}, c\boldsymbol{w} + c'\boldsymbol{w}'] = c[\boldsymbol{v},\boldsymbol{w}] + c'[\boldsymbol{v},\boldsymbol{w}'], \tag{A.19}$$

skew-symmetry:

$$[\boldsymbol{v},\boldsymbol{w}] = -[\boldsymbol{w},\boldsymbol{v}], \tag{A.20}$$

and Jacobi identity:

$$[\boldsymbol{u},[\boldsymbol{v},\boldsymbol{w}]] + [\boldsymbol{w},[\boldsymbol{u},\boldsymbol{v}]] + [\boldsymbol{v},[\boldsymbol{w},\boldsymbol{u}]] = 0. \tag{A.21}$$

The Lie bracket can be interpreted geometrically as the "infinitesimal commutator" of two one-parameter Lie group elements $\exp(t\boldsymbol{v})$ and $\exp(t\boldsymbol{w})$. When $\boldsymbol{v}$ and $\boldsymbol{w}$ are two vector fields in M and for each $x \in M$, the commuter:

$$\psi(t,\boldsymbol{x}) = \exp(-\sqrt{t}\boldsymbol{w})\exp(-\sqrt{t}\boldsymbol{v})\exp(\sqrt{t}\boldsymbol{w})\exp(\sqrt{t}\boldsymbol{v})\boldsymbol{x} \tag{A.22}$$

defines a curve for $t \geq 0$ sufficiently small. The vector defined by the Lie bracket $[\boldsymbol{v},\boldsymbol{w}]|_{\boldsymbol{x}}$ is the tangent vector to this curve at $\psi(0,\boldsymbol{x})$:

$$[\boldsymbol{v},\boldsymbol{w}] = \left.\frac{d}{dt}\right|_{t=0^+} \psi(t,\boldsymbol{x}). \tag{A.23}$$

It can be shown that these vector fields commute, that is, the order of the composition of their group actions does not matter:

$$\exp(t\boldsymbol{w})\exp(t\boldsymbol{v})x = \exp(t\boldsymbol{v})\exp(t\boldsymbol{w})x \tag{A.24}$$

if and only if

$$[\boldsymbol{v}, \boldsymbol{w}] = 0. \tag{A.25}$$

In local coordinates $\boldsymbol{x} = (x^1, \dots, x^{n_x})$, the Lie bracket operation commutes vector fields of the form (A.2), resulting in the following expression:

$$[\boldsymbol{v}, \boldsymbol{w}] = \boldsymbol{w}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} - \boldsymbol{v}\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}}$$

where the derivatives of the vector fields are represented by the Jacobian matrix of the vector fields $\boldsymbol{v}$ and $\boldsymbol{w}$.

In practical terms, commuting vector fields result in endpoints of a flow that are equal. This is desired in most applications where any kind of dynamics is to be rejected, which is not the case in process control. If one desires a stabilizing controller, commuting vector fields should be an objective secondary to reaching the tracking point or tracking zone, not a requirement.

A.6 Frobenius theorem and Lie algebras

Now that all the geometric objects have been approached, we seek ways to use them in process control. Frobenius' theorem establishes integrability conditions of a dynamic system composed of set of vector fields $\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_r\}$ on a smooth manifold M.

Theorem 5 (Frobenius) *The two following statements are equivalent:*

- *the system is in involution, or involutive;*
- *the system is integrable.*

The first statement can be proven if the following smooth real functions $h^k_{ij}(\boldsymbol{x})$, $\boldsymbol{x} \in M$, $i, j, k = 1, \dots, r$, such that for each $i, j = 1, \dots, r$, exist:

$$[\boldsymbol{v}_i, \boldsymbol{v}_j] = \sum_{k=1}^{r} h^k_{ij}\boldsymbol{v}_k \tag{A.26}$$

which means that every vector field generated by a bracket of two vector fields that belong to the manifold is also a vector field which belongs to the manifold. When this is true, the system is said to be involutive or in involution.

This condition is similar to the controllability matrix column rank condition for linear systems. As the system is proven integrable in a region, this region can be accessed by a specific configuration of the dynamic system vector fields.

Although this is a powerful theorem, it is computationally expensive to check if every vector field of a set can be generated by commuting two other vector fields. This is where the concept of *Lie algebra* is introduced.

Definition 4 *A Lie algebra is a vector space $\mathfrak{g}$ joined with the Lie bracket operation that satisfies the following axioms:*

- *Bilinearity:*

$$[c\boldsymbol{v}+c'\boldsymbol{v}',\boldsymbol{w}] = c[\boldsymbol{v},\boldsymbol{w}]+c'[\boldsymbol{v}',\boldsymbol{w}]$$
$$[\boldsymbol{v},c\boldsymbol{w}+c'\boldsymbol{w}',] = c[\boldsymbol{v},\boldsymbol{w}]+c'[\boldsymbol{v},\boldsymbol{w}']$$

- *Skew-symmetry:*

$$[\boldsymbol{v},\boldsymbol{w}] = -[\boldsymbol{w},\boldsymbol{v}]$$

- *Jacobi identity:*

$$[\boldsymbol{u},[\boldsymbol{v},\boldsymbol{w}]]+[\boldsymbol{w},[\boldsymbol{u},\boldsymbol{v}]]+[\boldsymbol{v},[\boldsymbol{w},\boldsymbol{u}]=0$$

for all $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{v}'$, $\boldsymbol{w}$, $\boldsymbol{w}'$ in $\mathfrak{g}$.

According to Olver (1993), it can be shown that the proof of integrability through evaluating if every bracket generated by two vector fields of the Lie group G can be reduced to evaluating the generators of the Lie algebra $\mathfrak{g}$.

If the generators are proven involutive in a manifold M then the dynamic system is completely integrable, according to Frobenius Theorem, and every point of the manifold is accessible by that dynamic system. The generators are, by definition, the smallest set of vector fields that belong to the algebra and are capable of generating the vector space $\mathfrak{g}$.

In process control it is not often that global properties of a dynamic system are required, since the process system itself shows several physical limitations, which limit the operational scope to a local set of states and inputs. With that in mind, the scope of analysis can be reduced to a local approach, resulting in a submanifold $N \in M$ with a local Lie group H and a local Lie algebra $\mathfrak{h}$ as well. In this case, the system is to be evaluated whether this submanifold where the states evolve is compatible with possible operating conditions.