UNIVERSITY OF SÃO PAULO
POLYTECHNIC SCHOOL

Rafael Ribeiro Sencio

Model predictive control based on the output prediction-oriented model: a dual-mode approach, and robust distributed algorithms

São Paulo

2022

Rafael Ribeiro Sencio

Model predictive control based on the output prediction-oriented model: a dual-mode approach, and robust distributed algorithms

Corrected Version

Thesis presented to the Polytechnic School of the University of São Paulo to obtain the degree of Doctor in Science.

Concentration area: Chemical Engineering

Advisor: Prof. Dr. Darci Odloak

São Paulo
2022

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

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, 8 de agosto de 2022

Assinatura do autor:

Assinatura do orientador:

Catalogação-na-publicação

Sencio, Rafael Ribeiro
Model predictive control based on the output prediction-oriented model: a dual-mode approach, and robust distributed algorithms / R. R. Sencio -- versão corr. -- São Paulo, 2022.
157 p.

Tese (Doutorado) - 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.

Rafael Ribeiro Sencio

Model predictive control based on the output prediction-oriented model: a dual-mode approach, and robust distributed algorithms

Thesis presented to the Polytechnic School of the University of São Paulo to obtain the degree of Doctor in Science.

Approved on June 9th 2022.

Examination Committee:

Prof. Dr. Darci Odloak
University of São Paulo

Prof. Dr. Julio Elias Normey Rico
Universidade Federal de Santa Catarina

Prof. Dr. Luís Cláudio Oliveira Lopes
Universidade Federal de Uberlândia

Prof. Dr. Márcio André Fernandes Martins
Universidade Federal da Bahia

Dr. Bruno Faccini Santoro
Op2B

To Luanna, my parents and my sister,
for their endless love, encouragement and support.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Prof. Darci Odloak for his thoughtful guidance, for his teachings, and for the opportunity to develop this research work.

I would also like to thank the defense committee, Professors Julio Normey Rico, Luís Cláudio Oliveira Lopes, Márcio Martins and Bruno Faccini Santoro, for their valuable comments and suggestions that helped me to improve the text.

I am extremely grateful to Guilherme for his constant review of my texts, for our fruitful exchange of experience in MATLAB, CasADi and LaTeX, and for the numerous conversations we had about control theory and the stability ingredients of different MPC formulations.

I would like to thank Prof. Thiago Costa for inviting me to collaborate on papers we published together during this period. I would also like to thank him for being the one who introduced me to this fascinating world of Model Predictive Control.

Thanks should also go to my friends from the Department of Telecommunications and Control Engineering, Fábio, Leonardo, Mateus and Gabriel, for our exchange of knowledge on control theory, and for the great time we spent together during our trip to take the LMI course at UNICAMP and during the CBA 2018. I am also grateful for always being welcome in their laboratory and for the espressos we had together at LCA.

Many thanks to my friends and colleagues of the B21 building: Antônio, Carminha, Daniel, Fábia, Gabriela, Guilherme, Gustavo Olivieri, Gustavo Taira, Lélia, Maria Giuliana, Ojonugwa, Patience, Patrícia, Prof. Guardani, Prof. Song, Regina, Renata Cavaignac, Renata Sousa, Ronald, Rubens, Steffany, Valter, Victtória and Zerwas. My apologies if I forgot to mention anyone. Thank you all so much for the many coffees we had together and for the great moments we shared.

I am also thankful to my parents and my sister for their support, patience, dedication, and for their belief in me.

Words cannot express my gratitude to Luanna, who strongly encouraged me to pursue this endeavor, always providing me with advice, emotional support, and love.

Finally, this research work would not have been possible without the financial support provided by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Finance Code 001, Process 8882.333509/2019-01.

"Nothing in life is to be feared, it is only to be understood.
Now is the time to understand more, so that we may fear less."
Marie Curie

Abstract

SENCIO, R. R. **Model predictive control based on the output prediction-oriented model: a dual-mode approach, and robust distributed algorithms**. 2022. 157p. Thesis (Doctor in Science) - Polytechnic School, University of São Paulo, São Paulo, 2022.

The output prediction-oriented model (OPOM) is a state-space model with incremental inputs that is derived from the analytical form of the system step response. This model has the prediction of the output at steady state as part of the state vector, which is useful for imposing the terminal constraint in an infinite horizon MPC formulation suitable for setpoint tracking. In this work, stabilizing MPC approaches based on the output prediction-oriented model are proposed, namely, a dual-mode MPC and three robust cooperative distributed MPC algorithms. First, a method to transform a state-space model with positional inputs into an OPOM-like model is presented. The resultant model can be viewed as a generalization of the traditional OPOM, being able to represent systems with stable, integrating and unstable poles as well as with pole multiplicity and time delays. Deploying a terminal control law and applying the concept of an invariant set for tracking, the proposed dual-mode MPC with OPOM has embedded integral action and guaranteed stability and feasibility under any setpoint change. In this approach, the characterization of steady outputs and inputs is based only on terminal states and inputs, avoiding parametrization of system equilibria. Such a method allows the computation of artificial references that are consistent with the true plant steady state, even in the presence of plant-model mismatch or unmeasured disturbances. The proposed MPC also addresses the case of output zone control and optimizing input targets. It is proved that, if the desired operating point is not admissible, the proposed controller steers the system to the operating point that minimizes an offset cost function. Concerning the proposed algorithms for robust cooperative distribute MPC, the multi-model system representation is adopted, and a plantwide performance index is optimized while imposing a cost-contracting constraint for all the models. This strategy results in a QCQP (quadratically constrained quadratic programming) problem that is solved for each subsystem. Local solutions are shared between the agents and an iterative procedure is applied to improve the overall solution. This approach is extended and two alternative algorithms are proposed. One is based on computing optimal weights of the convex combination of agents' solutions, which improves the convergence over iterations. The other enforces the robustness constraint only after iterations terminate, thereby turning the optimization problem solved by each agent into a QP (quadratic programming) problem. This strategy reduces the number of QCQP problems solved at each time step, which also reduces the CPU time spent by the agents.

The proposed distributed algorithms are suitable for dealing with both setpoint tracking and zone control problems, and have guaranteed recursive feasibility, convergence and stability. Numerical examples are presented to illustrate the application of the proposed controllers.

Keywords: Model predictive control. Robust distributed control. Cooperative distributed control. Setpoint tracking. Zone control.

Resumo

SENCIO, R. R. **Controle preditivo baseado no modelo orientado à predição da saída: uma abordagem dual e algoritmos distribuídos robustos**. 2022. 157p. Tese (Doutor em Ciências) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2022.

O modelo orientado à predição da saída (OPOM, no acrônimo em inglês) é um modelo de espaço de estado com entradas incrementais que é derivado da forma analítica da resposta ao degrau do sistema. Este modelo tem a previsão da saída em regime permanente como parte do vetor de estados, o que é útil para impor a restrição terminal em uma formulação de MPC de horizonte infinito adequada para rastreamento de referência. Neste trabalho, são propostas abordagens estabilizantes de MPC baseadas no modelo orientado à predição da saída, a saber, um MPC dual e três algoritmos de MPC distribuído cooperativo robusto. Primeiramente, é apresentado um método para transformar um modelo de espaço de estados com entradas posicionais em um modelo do tipo OPOM. O modelo resultante pode ser visto como uma generalização do OPOM tradicional, sendo capaz de representar sistemas com polos estáveis, integradores e instáveis, bem como com multiplicidade de polos e tempo morto. Utilizando uma lei de controle terminal e aplicando o conceito de conjunto invariante para rastreamento, o MPC dual com OPOM proposto possui ação integral embutida e garantia de estabilidade e viabilidade sob qualquer mudança de referência. Nesta abordagem, a caracterização de saídas e entradas em regime permanente é baseada apenas em estados e entradas terminais, evitando a parametrização do equilíbrio do sistema. Tal método permite o cálculo de referências artificiais que são consistentes com o verdadeiro estado estacionário da planta, mesmo na presença de incompatibilidade planta-modelo ou perturbações não medidas. O MPC proposto também aborda o caso de controle por zonas das saídas e alvos ótimos das entradas. Prova-se que, se o ponto de operação desejado não for admissível, o controlador proposto direciona o sistema para o ponto de operação que minimiza uma função de custo do desvio. Com relação aos algoritmos propostos para MPC distribuído cooperativo robusto, adota-se a representação multi-modelo do sistema, e otimiza-se um índice de desempenho de toda a planta enquanto uma restrição de contração de custo para todos os modelos é imposta. Essa estratégia resulta em um problema QCQP (acrônimo em inglês para programação quadrática com restrição quadrática) que é resolvido para cada subsistema. As soluções locais são compartilhadas entre os agentes e um procedimento iterativo é aplicado para melhorar a solução geral. Esta abordagem é estendida e dois algoritmos alternativos são propostos. Um é baseado no cálculo de pesos ótimos da combinação convexa das soluções dos agentes, o que melhora a convergência ao longo das iterações. O outro impõe a restrição de robustez somente após o término das iterações, transformando assim o problema de otimização resolvido por cada agente em um problema QP (acrônimo em inglês para programação quadrática). Essa estratégia reduz o número de problemas QCQP resolvidos a cada período de amostragem, o que também reduz o tempo de CPU gasto pelos agentes. Os algoritmos distribuídos propostos são adequados para lidar com problemas de rastreamento de referência e controle por zonas, e possuem garantia de viabilidade recursiva, convergência e estabilidade. Exemplos numéricos são apresentados para ilustrar a aplicação dos controladores propostos.

Palavras-chave: Controle preditivo. Controle distribuído robusto. Controle distribuído cooperativo. Rastreamento de referência. Controle por zonas.

List of Figures

List of Tables

List of Acronyms

CARIMA	Controlled Auto-Regressive, Integrated and Moving-Average
CARMA	Controlled Auto-Regressive and Moving-Average
CPU	Central Processing Unit
CSGPC	Constrained Stable Generalized Predictive Control
CSTR	Continuous Stirred-Tank Reactor
DMC	Dynamic Matrix Control
DMPC	Distributed Model Predictive Control
GPC	Generalized Predictive Control
IHMPC	Infinite Horizon Model Predictive Control
LDMC	Linear Dynamic Matrix Control
LMI	Linear Matrix Inequalities
LQR	Linear Quadratic Regulator
MIMO	Multiple-Input, Multiple-Output
MPC	Model Predictive Control
MPHC	Model Predictive Heuristic Control
NLP	Non-Linear Programming
OPOM	Output Prediction-Oriented Model
QCQP	Quadratically Constrained Quadratic Programming
QDMC	Quadratic Dynamic Matrix Control
QP	Quadratic Programming
RDMPC	Robust Distributed Model Predictive Control
RHC	Receding Horizon Control
RMPC	Robust Model Predictive Control
RTO	Real-Time Optimization

SDP	Semi-Definite Program
SISO	Single-Input, Single-Output
SMOC	Shell Multivariable Optimizing Controller
SOCP	Second-Order Cone Program

List of Symbols

Chapter 2

n_y	number of outputs
n_u	number of inputs
G	matrix of transfer functions
$G_{i,j}$	transfer function of output i w.r.t. input j
$S_{i,j}$	unit step response of output i w.r.t. input j
$d^0_{i,j}$	static coefficient of $S_{i,j}$
$d^d_{i,j,l}$	dynamic coefficient of $S_{i,j}$ related to the l-th pole
$r_{i,j,l}$	l-th continuous-time pole of $S_{i,j}$
T_s	sampling time
k	time step
F	dynamic-state transition matrix
D^d	matrix of dynamic coefficients $d^d_{i,j,l}$
D^0	matrix of static coefficients $d^0_{i,j}$
n_d	number of dynamic coefficients $d^d_{i,j,l}$
u	vector of inputs
Δu	vector of input increments
x_s	vector of artificial integrating states of the traditional OPOM
x_d	vector of dynamic states of the traditional OPOM
N	incidence matrix that defines which inputs affect each dynamic state
Ψ	output matrix related to x_d
x	vector of states
y	vector of outputs
A	state transition matrix

B	input matrix
C	output matrix
Δx	vector of state increments
y_s	vector of artificial integrating states of the new OPOM
B_s	input matrix related to y_s
B_d	input matrix related to x_d
n_{st}	number of stable poles
n_{in}	number of integrating poles
n_{un}	number of unstable poles
T	similarity transformation matrix
$\tilde{x}_{st}$	stable states of the decomposed system
$\tilde{x}_{in}$	integrating states of the decomposed system
$\tilde{x}_{un}$	unstable states of the decomposed system
$\tilde{A}_{st}$	state transition matrix related to $\tilde{x}_{st}$
$\tilde{A}_{in}$	state transition matrix related to $\tilde{x}_{in}$
$\tilde{A}_{un}$	state transition matrix related to $\tilde{x}_{un}$
$\tilde{B}_{st}$	input matrix related to $\tilde{x}_{st}$
$\tilde{B}_{in}$	input matrix related to $\tilde{x}_{in}$
$\tilde{B}_{un}$	input matrix related to $\tilde{x}_{un}$
$\tilde{C}_{st}$	output matrix related to $\tilde{x}_{st}$
$\tilde{C}_{in}$	output matrix related to $\tilde{x}_{in}$
$\tilde{C}_{un}$	output matrix related to $\tilde{x}_{un}$
x_{st}	vector of stable states of the new OPOM
x_{in}	vector of integrating states of the new OPOM
x_{un}	vector of unstable states of the new OPOM
A_{st}	state transition matrix related to x_{st}

A_{in}	state transition matrix related to x_{in}
A_{un}	state transition matrix related to x_{un}
B_{st}	input matrix related to x_{st}
B_{in}	input matrix related to x_{in}
B_{un}	input matrix related to x_{un}
C_{st}	output matrix related to x_{st}
C_{in}	output matrix related to x_{in}
C_{un}	output matrix related to x_{un}
$\tilde{x}$	state vector of the system with time delay
$\tilde{A}$	state transition matrix of the system with time delay
$\tilde{B}^{(i)}$	input matrix related to the inputs delayed by i time steps
q	the largest time delay between any output-input pair of the system
A_i	matrix of coefficients related to the outputs delayed by i time steps
B_i	matrix of coefficients related to the inputs delayed by i time steps

Chapter 3

n_{st}	number of stable poles
n_{in}	number of integrating poles
n_{un}	number of unstable poles
n_y	number of outputs
n_u	number of inputs
q	the largest time delay between any output-input pair of the system
x	vector of states
Δu	vector of input increments
y	vector of outputs
A	state transition matrix

B	input matrix
C	output matrix
y_s	vector of artificial integrating states of OPOM
x_{st}	vector of stable states of OPOM
x_{in}	vector of integrating states of OPOM
x_{un}	vector of unstable states of OPOM
A_{st}	state transition matrix related to x_{st}
A_{in}	state transition matrix related to x_{in}
A_{un}	state transition matrix related to x_{un}
$B_s^{(i)}$	input matrix related to y_s and to the inputs delayed by i time steps
$B_{st}^{(i)}$	input matrix related to x_{st} and to the inputs delayed by i time steps
$B_{in}^{(i)}$	input matrix related to x_{in} and to the inputs delayed by i time steps
$B_{un}^{(i)}$	input matrix related to x_{un} and to the inputs delayed by i time steps
C_{st}	output matrix related to x_{st}
C_{in}	output matrix related to x_{in}
C_{un}	output matrix related to x_{un}
x_d	vector that groups states x_{st}, x_{in} and x_{un} of OPOM
$\mathcal{Y}$	set of admissible outputs
$\mathcal{U}$	set of admissible inputs
$\mathcal{U}_\Delta$	set of admissible input increments
$\bar{y}_s$	steady output
$\bar{u}_s$	steady input
K	a stabilizing control gain such that $A_d + B_d K$ is discrete-time Hurwitz
F	closed-loop transition matrix of x_d given by $F = A_d + B_d K$
k	time step
Γ	matrix given by $\Gamma = (I - F)^{-1}$

w	augmented state given by $w = (x, u^-)$
A_w	closed-loop transition matrix of w
λ	approximation parameter
$\mathcal{W}_\lambda$	polyhedron defined by system constraints for a given λ
Ω_t^w	positively invariant set for tracking
$\bar{\mathcal{Y}}_s$	set of steady outputs that can be tracked without steady error
$\bar{\mathcal{Z}}_s$	set of admissible steady outputs and inputs that are consistent with Ω_t^w
$\bar{z}_s$	pair of steady outputs and inputs given by $\bar{z}_s = (\bar{y}_s, \bar{u}_s)$
N	control horizon length
$\mathbf{\Delta u}$	sequence of N input increments
y_{sp}	output setpoint
V	performance criterion
Q_y	matrix that penalizes the output error w.r.t. a given output reference
R	matrix that penalizes the input increments
V_N	performance criterion for a given control horizon length N
P	terminal weighting matrix
δ	slack variable
S_y	matrix that penalizes the steady-state output error
Q_u	matrix that penalizes the input error w.r.t. a given input reference
P_N	control problem defined for a given control horizon length N
$\mathcal{W}_N$	set of initial states w such that problem P_N is feasible
κ_N	implicit control law of the MPC controller based on P_N
$\mathbf{\Delta u}^*$	optimal sequence of N input increments
$\mathbf{\Delta \tilde{u}}$	suboptimal sequence of N input increments
V_N^*	optimal performance criterion
$\tilde{V}_N$	suboptimal performance criterion

$\mathcal{Y}_{sp}$	set of y_{sp} defined by the control zone
u_{sp}	input target
V_O	offset cost function
$\bar{z}_{sp}$	pair of output setpoint and input target given by $\bar{z}_{sp} = (\bar{y}_{sp}, \bar{u}_{sp})$
V_N^{zt}	performance criterion of the control problem with zone control and input targets
P_N^{zt}	control problem defined for the case of zone control and input targets for a given control horizon length N
κ_N^{zt}	implicit control law of the MPC controller based on P_N^{zt}
$\bar{\mathcal{Z}}_s^*$	set of all possible optimal operating points

Chapter 4

x	vector of plantwide states
u	vector of plantwide input
y	vector of plantwide outputs
n_x	number of plantwide states
n_u	number of plantwide inputs
n_y	number of plantwide outputs
A	plantwide state transition matrix
B	plantwide input matrix
C	plantwide output matrix
y_s	vector of artificial integrating states of the new OPOM
x_d	vector of dynamic states of the new OPOM
Δu	vector of input increments
F	state transition matrix related to x_d
B_s	input matrix related to y_s
B_d	input matrix related to x_d

Ψ	output matrix related to x_d
M	number of subsystems
x_i	vector of states of subsystem i
u_i	vector of input of subsystem i
y_i	vector of outputs of subsystem i
A_i	state transition matrix of subsystem i
B_{ij}	matrix that describes the effects of inputs of subsystem j on the states of subsystem i
C_i	output matrix of subsystem i
n_{x_i}	number of states of subsystem i
n_{u_i}	number of inputs of subsystem i
n_{y_i}	number of outputs of subsystem i
B_i	matrix that describes the effects of inputs of subsystem i on the plantwide states
$B_{s,i}$	matrix that describes the effects of inputs of subsystem i on the plantwide states y_s
$B_{d,i}$	matrix that describes the effects of inputs of subsystem i on the plantwide states x_d
Δu_i	vector of input increments of subsystem i
$\mathcal{U}$	set of admissible inputs
$\mathcal{U}_\Delta$	set of admissible input increments
y_{sp}	output setpoint
$\mathcal{Y}_{sp}$	set of y_{sp} defined by the control zone
N	control horizon length
$\Delta \mathbf{u}_i$	sequence of N input increments of subsystem i
x_{sp}	OPOM state setpoint given by $x_{sp} = (y_{sp}, 0)$
V	plantwide performance criterion

$\mathbf{\Delta u}$	sequence of N plantwide input increments
Q	matrix that penalizes the state error w.r.t. a given state reference
R	matrix that penalizes the input increments
Q_s	submatrix of Q
Q_d	submatrix of Q
Q_{sd}	submatrix of Q
P	terminal weighting matrix
δ	slack variable that corresponds to the steady-state output error
S	matrix that penalizes the steady-state output error
δ_x	slack variable given by $\delta_x = (\delta, 0)$
x_s	state artificial reference
θ	model with matrices A_θ, B_θ and C_θ
Ω	set of models θ
L	number of models θ in Ω
θ_n	nominal model
θ_t	true plant model
V_θ	plantwide performance criterion computed for model θ
x_θ	vector of plantwide states of model θ
$x_{s,\theta}$	state artificial reference of model θ
$x_{d,\theta}$	vector of plantwide states $x_{d,\theta}$ of model θ
$y_{s,\theta}$	vector of plantwide states $y_{s,\theta}$ of model θ
P_θ	terminal weighting matrix of model θ
F_θ	state transition matrix related to $x_{d,\theta}$
$(\mathbf{\Delta u}^{[0]}, y_{sp}^{[0]})$	initial condition of the distributed algorithm
p	iteration
P_i	control problem solved by agent i

$(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$	plantwide solution obtained at iteration p
$\Delta\mathbf{u}_i^*$	vector of optimal input increments of subsystem i computed by agent i
$y_{sp,i}^*$	optimal plantwide output setpoint computed by agent i
w_i	weight of subsystem i in the convex combination of solutions
$\Delta\bar{\mathbf{u}}_i^*$	vector $\Delta\mathbf{u}^{[p-1]}$ partially updated with $\Delta\mathbf{u}_i^*$
$\bar{p}$	iteration at which the iterative procedure terminates
k	time step
κ_N	implicit control law of the distributed MPC controller
$\Delta\mathbf{u}^*$	optimal sequence of N plantwide input increments
y_{sp}^*	optimal plantwide output setpoint
V^*	optimal plantwide performance criterion
$\Delta\tilde{\mathbf{u}}$	suboptimal sequence of N plantwide input increments
$\tilde{V}$	suboptimal plantwide performance criterion
$\tilde{y}_{sp}$	suboptimal plantwide output setpoint
P_w	optimization problem that computes optimal weights of the convex combination
P_i^{QP}	control problem solved by agent i in Algorithm 3

Contents

Chapter 1

Introduction

1.1 Model predictive control technology

Model predictive control (MPC) is a generic term that refers to a class of control algorithms that explicitly use a system model to compute and optimize the open-loop plant response over a prediction horizon. This future system behavior is usually penalized in terms of the deviation from a given reference in the definition of a performance index, also known as cost function. The objective is to optimize this performance index by solving an optimization problem, possibly subject to constraints. The decision variables of the optimal control problem usually consist of a sequence of system inputs or input increments over a control horizon. Although a sequence of future control actions is obtained at each time instant, only the first input, i.e., the one corresponding to the current time step, is applied to the system. Then, once new measurements are available at the next sampling time, the MPC problem is solved again, which implicitly defines a feedback control law. This procedure is called moving or receding horizon principle, which is the reason why MPC is also known as receding horizon control (RHC) (MUSKE; RAWLINGS, 1993).

Considered a well-established and accepted control strategy, MPC has attracted significant interest from both industry and academia over the last few decades. Such success can be mostly explained by advantageous features that MPC possesses. For instance, MPC is among the few control strategies that systematically deal with system constraints, e.g., actuator limits and process variable specifications. This allows the plant to safely achieve economically optimal operating points typically located at the intersection of constraints (QIN; BADGWELL, 2003). Besides considering important system constraints, another attractive aspect of MPC is the optimization of a user-specified performance index that can be easily defined by practitioners through physically meaningful metrics and tuning parameters. In addition to reflecting control purposes, the cost function can also incorporate economic objectives in order to obtain a more profitable operation, even during transients. Moreover, in the MPC algorithm, feedforward information and system time delays can be explicitly taken into account over predictions through the use of a plant model. In general, these models are obtained from plant tests, not requiring advanced methods of system identification. Since models that describe MIMO (multiple-input, multiple-output) systems can be employed, this gives MPC the ability to straightforwardly handle multivariable systems.

Early developments and industrial applications of MPC controllers date from the end of the 1970s. The first implementations reported in the literature are the Model Predictive Heuristic Control (MPHC), also known as IDCOM and developed by Richalet *et al.* (1978), and the Dynamic Matrix Control (DMC), proposed by Shell engineers Cutler and Ramaker (1980). Improvements in the DMC algorithm have been made towards the inclusion of input and output constraints, as presented in the Linear Dynamic Matrix Control (LDMC) (MORSHEDI; CUTLER; SKROVANEK, 1985), and the use of a quadratic cost function, deployed in the Quadratic Dynamic Matrix Control (QDMC) (GARCIA; MORSHEDI, 1986).

Even though MPC applications had already been consolidated in the late 1980s, there were unresolved issues to be addressed, such as robustness and stability of the available algorithms. In this regard, Garcia and Morari (1982) proposed a controller design method for SISO (single-input, single-output) systems that considers explicitly the uncertainty caused by plant modeling errors and provided practical tuning guidelines. An extension of this technique was presented in Garcia and Morari (1985a) for MIMO systems, while the control law computation, as well as more tuning guidelines, were discussed in Garcia and Morari (1985b). Other works that address robustness and stability problems of MPC controllers may be highlighted such as Rouhani and Mehra (1982) and Morari and Zafiriou (1989).

All the techniques described above use discrete step or impulse response data rather than parametric models. However, Clarke, Mohtadi and Tuffs (1987a) and Clarke, Mohtadi and Tuffs (1987b) proposed the Generalized Predictive Control (GPC), which is a parametric approach based on CARMA (Controlled Auto-Regressive and Moving-Average) or CARIMA (Controlled Auto-Regressive, Integrated and Moving-Average) models. These approaches have encouraged the development of other parametric model-based MPC controllers, such as the state-space formulation proposed in Li, Lim and Fisher (1989). At the same time, Shell engineers developed the Shell Multivariable Optimizing Controller (SMOC), described as a bridge between state-space models and MPC algorithms (MARQUIS; BROUSTAIL, 1988; YOUSFI; TOURNIER, 1991).

A comprehensive historical development of the industrial MPC technology and a summary of the main commercial solutions based on this class of algorithms can be found in the survey of Qin and Badgwell (2003). More recent industrial MPC products and implementation steps are detailed in Lahiri (2017).

The use of state-space models not only allowed the application of MPC to a larger class of systems, i.e. with stable, integrating and unstable dynamics, but it also paved the way for the development of new MPC formulations with strong stabilizing properties. Ingredients needed for ensuring stability of MPC controllers are well discussed in Mayne *et al.* (2000) and in Mayne and Falugi (2019), while theoretical aspects of different MPC formulations can be found in tutorial papers (RAWLINGS, 2000; WANG, 2004) as well as in many textbooks (MACIEJOWSKI, 2002; ROSSITER, 2004; CAMACHO; BORDONS, 2007; RAWLINGS; MAYNE; DIEHL, 2017).

Although today MPC can be considered a well-matured control strategy, it is still a matter of research and several contributions continue to appear in the literature every year.

1.2 Motivation

The analysis and design of MPC using state-space models directly benefits from optimal control theory, which explains why the majority of the modern MPC literature has been developed using this model framework. Despite this fact, few commercial MPC packages support state-space models (QIN; BADGWELL, 2003) and, currently, the most popular industrial MPC vendors still employ step/impulse response models and transfer functions (KANO; OGAWA, 2010; WHEAT, 2018; FERNANDES *et al.*, 2020).

As a new interpretation of DMC-like controllers, an approach that rearranges step/impulse response models into a state-space form has been proposed by Li, Lim and Fisher (1989). However, since the resulting state vector has the dimension of the number of step response coefficients times the number of outputs, the proposed method produces a state-space model with a potentially very large order.

With the purpose of representing DMC-like controllers with fewer states, Gouvêa and Odloak (1997) proposed a reduced-order state-space model with incremental inputs and whose state vector contains the prediction of system output at steady state. Thus, unlike DMC state predictors that often need a large number of coefficients to capture plant steady state, the approach of Gouvêa and Odloak (1997) deploys a much more parsimonious model. This model has been used in an MPC that addresses robust stability via linear-matrix inequalities (LMI) methods (RODRIGUES; ODLOAK, 2000). Another relevant advantage of the state-space model proposed by Gouvêa and Odloak (1997) is revealed in (RODRIGUES; ODLOAK, 2003a), in which the output prediction at steady state is shown to be a useful feature for establishing the terminal constraint in an infinite horizon MPC (IHMPC). In this work of Rodrigues and Odloak (2003b), the model proposed by Gouvêa and Odloak (1997) was rearranged and named as the output prediction-oriented model (OPOM), as it has been known ever since.

Several MPC formulations based on the OPOM model have been proposed in the literature (RODRIGUES; ODLOAK, 2003a; ODLOAK, 2004; CARRAPIÇO; ODLOAK, 2005; SANTORO; ODLOAK, 2012; MARTINS *et al.*, 2013; MARTINS; ODLOAK, 2016). These studies have led to successful practical applications in laboratory-scale plants (MARTIN; ODLOAK; KASSAB, 2013; SILVA *et al.*, 2020) and even in oil refineries (PORFÍRIO; ALMEIDA NETO; ODLOAK, 2003; CARRAPIÇO *et al.*, 2009; MARTIN; ZANIN; ODLOAK, 2019). The MPC with OPOM is now part of an in-house advanced control package developed by Petrobras (Petróleo Brasileiro S.A.) and has been implemented in many other process units of the main oil refineries of Brazil.

Concerning the literature of MPC strategies based on the OPOM model, the following open issues served as motivation to the research developed in this work:

- Naturally, the OPOM model has been incrementally improved and extended formulations have been proposed for dealing with different types of system dynamics. However, this often leads to *ad hoc* rules regarding the way OPOM is built, which not only divides the OPOM formulation into many cases but also makes model construction less straightforward. Therefore, an easier and more general method to build OPOM is still lacking.

- Concerning the infinite horizon MPC with OPOM, stability is guaranteed by forcing to zero terminal states related to integrating and unstable manifolds. This is usually performed by imposing a terminal equality constraint, which can be very restrictive and, upon conflicts with input constraints, leads to problem infeasibilities. To avoid this, the available formulations of IHMPC with OPOM often employ slack variables to soften this equality constraint, thereby ensuring problem feasibility. The downside of this procedure is evident when dealing with integrating and unstable systems, for which stability guarantee is only recovered when slack variables are zeroed. An alternative way of avoiding the terminal equality constraint consists of employing a terminal inequality constraint instead. This constraint forces the terminal state into a (non-trivial) invariant set in which a terminal control law is considered. Such an approach has not been applied to the IHMPC with OPOM yet.

- In large-scale plants, distributing the controller while exchanging information between subsystems can be advantageous not only due to the reduction of local computational requirements but mainly because it facilitates organizational aspects such as commissioning and maintenance. To address such cases, distributed MPC formulations based on OPOM have recently been proposed (SANTANA; MARTINS; ODLOAK, 2020b; SARAPKA; MARTINS; ODLOAK, 2021). These approaches deploy a cooperative algorithm, i.e., subsystems pursue a common objective and thus optimize a plantwide performance index. However, since these controllers are based on a single linear model, their performance can deteriorate due to changing economic criteria that can lead the system to operate far from the steady state in which the model was obtained. Therefore, existing robust MPC strategies based on OPOM (ODLOAK, 2004) can be extended for dealing with distributed systems, which represents a viable and important contribution to enhance robustness while guaranteeing system stability.

1.3 Objectives

Motivated by the aforementioned aspects, the main objective of this thesis is to propose:

- a unified and simple method to build OPOM;
- a general infinite horizon MPC with OPOM that deploys a terminal control law (dual-mode approach);
- a robust cooperative distributed MPC with OPOM.

Along with pertinent literature review, these objectives are better supported and justified in the subsequent chapters, whose content will be outlined in the sequel.

1.4 Outline of the thesis

This thesis is organized into five chapters, including the present introductory one. The remaining chapters are outlined as follows.

In Chapter 2, a straightforward method for building an OPOM-like model is proposed. The new OPOM, as it is called here, is first proposed for open-loop stable systems and then extended for the more general case considering systems with stable, integrating and unstable poles as well as with time delays. A comparison of the new OPOM with the traditional one is given. It is also shown how the state vector of the new OPOM can be computed to cope with output feedback. The advantages of the new OPOM are highlighted before finishing the chapter.

Chapter 3 first addresses the characterization of artificial references as well as the definition of the invariant set for tracking. Then, the formulation of a dual-mode MPC with OPOM for tracking setpoints is presented along with analyses regarding the controller's recursive feasibility and closed-loop convergence. The proposed MPC is applied to a double-integrator system. An extension of the dual-mode MPC with OPOM for handling output control zones and input targets is presented and technical lemmas and theorem are proven. Finally, numerical results regarding the application of the proposed strategy to an unstable reactor system are given.

In Chapter 4, a robust cooperative distributed MPC based on the OPOM model that handles the zone control problem is proposed. Technical lemmas and theorem regarding algorithm convergence, recursive feasibility and closed-loop stability are presented. The proposed algorithm is applied to a high-purity distillation column and then compared with a centralized robust MPC and a nominal distributed MPC. Then, two alternative algorithms are proposed, namely, one that improves the convergence of subsystems' solutions over iterations and another that robustifies a nominal cooperative distributed MPC. Theoretical analyses are presented. Numerical results in a simulated two reactors-separator system are presented along with the comparison of CPU (central processing unit) times spent by the proposed strategies. The first two algorithms proposed in Chapter 4 have been published in Sencio and Odloak (2022).

Finally, Chapter 5 serves as an overview of the main contributions of this thesis and presents directions for future research.

1.5 Notation

For a given positive-definite symmetric matrix $P \in \mathbb{R}^{n\times n}$, the weighted Euclidean norm of $x \in \mathbb{R}^n$ is denoted as $\|x\|_P := \sqrt{x^T P x}$. Matrix $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix. Besides denoting the number zero, the symbol 0 also represents a matrix whose entries are zeros. When not indicated, matrix dimension is determined by the context. Given two matrices $M,\ N \in \mathbb{R}^{n\times n}$, $M > N$ denotes that $(M - N)$ is a positive-definite matrix. For a matrix $M \in \mathbb{R}^{n\times n}$, $\lambda_{\min}(M)$ denotes the smallest eigenvalue of M. Given two vectors $a,\ b \in \mathbb{R}^n$, $a \leq b$ corresponds to an element-wise inequality. A vector $[a^T\, b^T]^T$ is simply represented by (a, b). Given integers a and b, with $b > a$, the set $\mathbb{I}_{a:b}$ is defined as $\mathbb{I}_{a:b} := \{a, a+1, \ldots, b-1, b\}$, while $\mathbb{I}_{\geq a}$ denotes the set of integers greater than or equal to a. For $a \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and set $\Psi \subset \mathbb{R}^{n+m}$, $\text{Proj}_a(\Psi) := \{a \in \mathbb{R}^n : \exists b \in \mathbb{R}^m, (a, b) \in \Psi\}$ denotes the projection operation. For an initial state $x(k)$, $x(j|k)$ represents the j-steps ahead predicted state and $\Delta u(j|k)$ denotes the j-th control increment of a sequence $\Delta \mathbf{u}_k$ computed at time step k, in which j and k are integers. To simplify notation, the indication of time instant k is omitted when its use is not necessary, with x^+ and u^- denoting the successor state and the predecessor input, respectively. Other definitions will be presented throughout this thesis as needed. Although the general notation is the same, some symbols have different meanings in each chapter.

Chapter 2

The new output prediction-oriented model

2.1 Introduction

The output prediction-oriented model is a state-space model that is written in the incremental form of inputs and that is derived from the analytical form of the step response of the system (RODRIGUES; ODLOAK, 2003b). A particular feature of this type of model is that it has the output steady-state prediction as one of the states, which is useful for imposing the end constraint in the infinite horizon MPC. Enjoying this property, many IHMPC formulations based on the OPOM model have been proposed considering systems with stable (RODRIGUES; ODLOAK, 2003b; ODLOAK, 2004), integrating (RODRIGUES; ODLOAK, 2003a; CARRAPIÇO; ODLOAK, 2005) and unstable (MARTINS; ODLOAK, 2016) poles as well as with time delay (SANTORO; ODLOAK, 2012; MARTINS *et al.*, 2013). Practical applications of MPC based on OPOM have been reported in the literature (PORFÍRIO; ALMEIDA NETO; ODLOAK, 2003; CARRAPIÇO *et al.*, 2009; MARTIN; ODLOAK; KASSAB, 2013; MARTIN; ZANIN; ODLOAK, 2019; SILVA *et al.*, 2020).

To exemplify the construction of the OPOM model, let us consider a simple case of a MIMO open-loop stable system with n_y outputs and n_u inputs described by the following matrix of transfer functions:

$$G(s) = \begin{bmatrix} G_{1,1}(s) & \cdots & G_{1,n_u}(s) \\ \vdots & \ddots & \vdots \\ G_{n_y,1}(s) & \cdots & G_{n_y,n_u}(s) \end{bmatrix} \tag{2.1}$$

in which each transfer function $G_{i,j}(s)$ describes the relationship between output i and input j and is given as follows:

$$G_{i,j}(s) = \frac{b_{i,j,0} + b_{i,j,1}s + b_{i,j,2}s^2 + \cdots + b_{i,j,n_b}s^{n_b}}{1 + a_{i,j,1}s + a_{i,j,2}s^2 + \cdots + a_{i,j,n_a}s^{n_a}}, \quad n_a,\ n_b \in \mathbb{N},\ n_b < n_a$$

After performing a partial fraction expansion, the unit step response of $G_{i,j}(s)$ is:

$$S_{i,j}(s) = \frac{G_{i,j}(s)}{s} = \frac{d^0_{i,j}}{s} + \frac{d^d_{i,j,1}}{s - r_{i,j,1}} + \dots + \frac{d^d_{i,j,n_a}}{s - r_{i,j,n_a}} \tag{2.2}$$

where $d^0_{i,j}$ and $d^d_{i,j,l}$, $l \in \mathbb{I}_{1:n_a}$, are static and dynamic coefficients, respectively, and $r_{i,j,l}$, $l \in \mathbb{I}_{1:n_a}$ are the system poles, assumed distinct. By taking the inverse Laplace transform of (2.2) and considering a sampling time T_s, the corresponding discretized unit step response is:

$$S_{i,j}(k) = d^0_{i,j} + \sum_{l=1}^{n_a} d^d_{i,j,l} e^{r_{i,j,l}kT_s}, \quad \forall k \in \mathbb{I}_{\geq 0} \tag{2.3}$$

Now, let us define the following matrices:

$$\begin{aligned}
F = \operatorname{diag}\Big(& e^{r_{1,1,1}T_s}, \dots, e^{r_{1,1,n_a}T_s}, \dots, e^{r_{1,n_u,1}T_s}, \dots, e^{r_{1,n_u,n_a}T_s}, \dots, \\
& e^{r_{n_y,1,1}T_s}, \dots, e^{r_{n_y,1,n_a}T_s}, \dots, e^{r_{n_y,n_u,1}T_s}, \dots, e^{r_{n_y,n_u,n_a}T_s}\Big) \\
D^d = \operatorname{diag}\Big(& d^d_{1,1,1}, \dots, d^d_{1,1,n_a}, \dots, d^d_{1,n_u,1}, \dots, d^d_{1,n_u,n_a}, \dots, \\
& d^d_{n_y,1,1}, \dots, d^d_{n_y,1,n_a}, \dots, d^d_{n_y,n_u,1}, \dots, d^d_{n_y,n_u,n_a}\Big) \\
D^0 = & \begin{bmatrix} d^0_{1,1} & \cdots & d^0_{1,n_u} \\ \vdots & \ddots & \vdots \\ d^0_{n_y,1} & \cdots & d^0_{n_y,n_u} \end{bmatrix}
\end{aligned}$$

in which $F \in \mathbb{C}^{n_d \times n_d}$, $D^d \in \mathbb{C}^{n_d \times n_d}$, $D^0 \in \mathbb{R}^{n_y \times n_u}$ and $n_d = n_u n_a n_y$. Defining the incremental input as $\Delta u(k) = u(k) - u(k-1)$ and applying the superposition principle, the following state-space model is obtained:

$$\begin{bmatrix} x_s \\ x_d \end{bmatrix}^+ = \begin{bmatrix} I_{n_y} & 0 \\ 0 & F \end{bmatrix} \begin{bmatrix} x_s \\ x_d \end{bmatrix} + \begin{bmatrix} D^0 \\ D^d F N \end{bmatrix} \Delta u \tag{2.4}$$

$$y = \begin{bmatrix} I_{n_y} & \Psi \end{bmatrix} \begin{bmatrix} x_s \\ x_d \end{bmatrix} \tag{2.5}$$

in which $x_s \in \mathbb{R}^{n_y}$ and $x_d \in \mathbb{R}^{n_d}$ are system states and $N \in \mathbb{R}^{n_d \times n_u}$ and $\Psi \in \mathbb{R}^{n_y \times n_d}$ are incidence matrices given as follows:

$$N = \begin{bmatrix} J_1 \\ \vdots \\ J_{n_y} \end{bmatrix}, \quad J_i = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ & \vdots & & \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \; J_i \in \mathbb{R}^{n_u n_a \times n_u},$$

$$\Psi = \begin{bmatrix} \Phi & & 0 \\ & \ddots & \\ 0 & & \Phi \end{bmatrix}, \quad \Phi = \begin{bmatrix} 1 \cdots 1 \end{bmatrix}, \; \Phi \in \mathbb{R}^{n_u n_a}.$$

In the OPOM model defined in (2.4)-(2.5), x_s is the integrating state that appears when writing the model in the incremental form of inputs and x_d is related to the stable poles of the system. As a unique feature of the OPOM, the state x_s also corresponds to the prediction of the system output at steady-state. Note that, beyond the way OPOM is obtained, which is closely related to step response models often used by DMC controllers, the major difference between OPOM and other state-space models with incremental inputs (also called velocity-form models) is the embedded prediction of steady outputs. This feature, as already mentioned, is useful for stating the end constraint in an offset-free infinite horizon MPC formulation with guaranteed stability.

Although we here used the OPOM formulation presented by Odloak (2004), the underlying rationale regarding the construction of this type of model can be found in Rodrigues and Odloak (2003b) for open-loop stable systems. For other types of systems, the reader is referred to the works already cited at the beginning of this section.

The main steps needed to build the OPOM model from system transfer functions can be summarized as follows:

1. apply a unit step change to system inputs;
2. obtain the analytical form of the step response by partial fractions expansion;
3. define the states related to the incremental form of the inputs;
4. define the states related to the system poles (stable, integrating, or unstable);
5. build up the state-space model.

Even though the overall steps to build OPOM are the same, there exist many *ad hoc* rules depending on whether the system has stable, integrating, unstable poles, or time delay. For instance, if the system has complex poles, some OPOM matrices end up having entries with complex numbers, which may be undesirable, especially in applications that require solving an LMI problem, whose solvers cannot directly handle complex-valued matrices. In such cases, a different procedure must be applied, which involves using Euler's formula to obtain an OPOM model with real-valued states. Thus, due to these particularities for each system, it may be difficult to automate the OPOM construction for the general case in a computer program without using different subroutines for each type of system, e.g. stable and integrating, stable and unstable with time delay, etc. Furthermore, different model structures often lead to distinct MPC formulations, which may be an issue for building a more general industrial package.

Another aspect of the OPOM construction concerns the resulting model dimension. Obviously, any procedure that augments the states to obtain inputs in the incremental form results in a non-minimal state-space model. However, apart from the inevitable additional integrating states needed to obtain incremental inputs, the way that OPOM is built may produce unnecessarily large states x_d. This occurs because coefficients are packed in matrices F and D^d in a way that is oriented through each output-input pair, which does not economically deal with cases of pole multiplicity or when some poles are shared between different outputs.

Therefore, aiming at addressing the above-mentioned OPOM limitations, the objective of this chapter is to propose an alternative method for building up this type of model. Here, instead of starting from the transfer functions that define the dynamics between each input and output, we look for a general formulation that can be used for transforming a state-space model in the positional form of inputs into an OPOM-like model. Also, the proposed method deals with systems that have pole multiplicity and is easy to automate in a computer program, which is an advantage for developing an industrial package. Since the resulting state-space model preserves the same features as the original OPOM proposed in Rodrigues and Odloak (2003b), namely the prediction of the output at steady state as one of the states, it can also be considered as an output-prediction oriented model. However, to differentiate from the traditional OPOM, the proposed model will be called the new OPOM hereafter.

This chapter is organized as follows. In the next section, the method for building the new OPOM for open-loop stable systems is described and an example that illustrates the dynamics of this model is presented. Then, in Section 2.3 the proposed method is extended for systems with stable, integrating and unstable poles and an alternative formulation in which the dynamics of integrating states do not depend on system parameters is proposed. The case of time-delay systems is considered in Section 2.4 along with a numerical example that compares system responses produced by the new OPOM and the traditional one. Section 2.5 discusses the use of the realigned model as a starting model to build the new OPOM in order to obtain output feedback. Finally, Section 2.6 summarizes the advantages of the new OPOM over the traditional one and concludes the chapter.

2.2 The new OPOM for open-loop stable systems

The proposed method to build the new OPOM requires the system discrete-time state-space model, which in general can be easily obtained by linearization and discretization of rigorous differential equations models or even through system identification techniques, when experimental data are available. Although the steps described in the sequel are somewhat exhaustive, the final formulation is straightforward to be applied.

Consider the state-space model in the positional form of inputs defined below, in which $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$ and n_x, n_u and n_y are the number of states, inputs and outputs, respectively:

$$x^+ = Ax + Bu \tag{2.6}$$

$$y = Cx \tag{2.7}$$

For the sake of simplicity, we first assume here that the system (2.6)-(2.7) has only stable poles. An extension to a more general case will be provided in the next section.

Defining the incremental states $\Delta x(k) = x(k) - x(k-1)$, the following model can be obtained from (2.6) and (2.7):

$$\begin{aligned} \Delta x^+ &= A\Delta x + B\Delta u \\ y^+ &= y + C\Delta x^+ \end{aligned}$$

Suppose a step input with amplitude Δu is applied to the system at time instant k. The evolution of the system output can be given as follows:

$$\begin{aligned} y(k+1) &= y(k) + C\Delta x(k+1) \\ y(k+2) &= y(k+1) + C\Delta x(k+2) \\ &= y(k) + C\left(I + A\right)\Delta x(k+1) \\ &\vdots \\ y(k+j) &= y(k) + C\left(I + A + \cdots + A^{j-2} + A^{j-1}\right)\Delta x(k+1) \end{aligned}$$

Since we have assumed that A is stable, then the inverse $(I-A)^{-1}$ exists and the following expression holds:

$$S_j = I + A + \cdots + A^{j-2} + A^{j-1} = \sum_{i=0}^{j-1} A^i = \left(I - A^j\right)\left(I - A\right)^{-1} \tag{2.8}$$

The equality (2.8) can be easily verified by computing the difference $S_j - S_j A = I - A^j$. Now, the expression for $y(k+j)$ can be rewritten in such a way that we can separate the static part, which does not depend on j, from the dynamic one, which in turn depends on j:

$$\begin{aligned} y(k+j) &= y(k) + C\left(I - A^j\right)\left(I - A\right)^{-1}\left(A\Delta x(k) + B\Delta u(k)\right) \\ &= \underbrace{y(k) + C\left(I - A\right)^{-1}\left(A\Delta x(k) + B\Delta u(k)\right)}_{\text{static part}} \\ &\quad - \underbrace{CA^j\left(I - A\right)^{-1}\left(A\Delta x(k) + B\Delta u(k)\right)}_{\text{dynamic part}} \end{aligned} \tag{2.9}$$

Let us define the following states:

$$\begin{aligned} y_s(k) &= y(k) + C\left(I - A\right)^{-1} A\Delta x(k) \\ x_d(k) &= -\left(I - A\right)^{-1} A\Delta x(k) \end{aligned}$$

Remark 1. At steady state, observe that $\Delta x = 0$, which implies that $x_d = 0$ and $y_s = y$.

Then, (2.9) can be rewritten as follows:

$$y(k+j) = \underbrace{y_s(k) + C\left(I - A\right)^{-1} B\Delta u(k)}_{\text{static part}} + \underbrace{CA^j\left(x_d(k) - \left(I - A\right)^{-1} B\Delta u(k)\right)}_{\text{dynamic part}} \tag{2.10}$$

The substitution of $j = 0$ in (2.10) yields:

$$\begin{aligned} y(k) &= y_s(k) + C\left(I - A\right)^{-1} B\Delta u(k) + C\left(x_d(k) - \left(I - A\right)^{-1} B\Delta u(k)\right) \\ &= y_s(k) + Cx_d(k) \end{aligned} \tag{2.11}$$

Also, for $j = 1$, we have that:

$$y(k+1) = y_s(k) + C\left(I - A\right)^{-1} B\Delta u(k) + CA\left(x_d(k) - \left(I - A\right)^{-1} B\Delta u(k)\right) \tag{2.12}$$

Observe that, for $k \leftarrow k+1$, (2.11) also implies that $y(k+1) = y_s(k+1) + Cx_d(k+1)$. Then, from (2.12), the following expressions for $y_s(k+1)$ and $x_d(k+1)$ can be obtained:

$$\begin{aligned} y_s(k+1) &= y_s(k) + C\left(I - A\right)^{-1} B\Delta u(k) \\ x_d(k+1) &= Ax_d(k) - A\left(I - A\right)^{-1} B\Delta u(k) \end{aligned}$$

Therefore, the following OPOM-like model is obtained in the matrix form:

$$\begin{bmatrix} y_s \\ x_d \end{bmatrix}^+ = \begin{bmatrix} I_{n_y} & 0 \\ 0 & F \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} + \begin{bmatrix} B_s \\ B_d \end{bmatrix} \Delta u \tag{2.13}$$

$$y = \begin{bmatrix} I_{n_y} & \Psi \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} \tag{2.14}$$

in which

$$F = A, \qquad B_s = C\left(I - A\right)^{-1} B, \qquad B_d = -A\left(I - A\right)^{-1} B, \qquad \Psi = C.$$

In this model, y_s represents integrating states that correspond to the prediction of steady outputs and x_d is the system stable dynamic part. Thus, it has the same model structure as the one proposed in Rodrigues and Odloak (2003b), apart from the fact that the steady output is denoted here as y_s instead of x_s. In fact, the above presented method generates exactly the same OPOM model if the original system given in (2.6) and (2.7) has the following properties:

- the original model is not minimal in the case some outputs share states in common;
- C is a binary matrix;
- A is a diagonal matrix.

Particularly, if A is a diagonal matrix, the proposed method generates a model that inherits an interesting property of the traditional OPOM model, which is the possibility of explicitly solving the Lyapunov equation. For instance, $P - A^T P A = Q$ can be easily solved as $P = \left(I - A^T A\right)^{-1} Q$ when A is a diagonal matrix.

In the sequel, we provide an example to illustrate the dynamics of the new OPOM model.

2.2.1 Example

Consider a SISO second-order non-minimal phase system (MUSKE; RAWLINGS, 1993) written in the form of (2.6)-(2.7) with the following matrices:

$$A = \begin{bmatrix} 4/3 & -2/3 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} -2/3 & 1 \end{bmatrix}$$

Using the final formulation of the proposed method given in (2.13) and (2.14), then the resulting OPOM model of this system can be written as follows:

$$\begin{bmatrix} y_s \\ x_d \end{bmatrix}^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4/3 & -2/3 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ -3 \end{bmatrix} \Delta u$$

$$y = \begin{bmatrix} 1 & -2/3 & 1 \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix}$$

Figure 1 shows the system response with the application of unit-step input at time instants $k = 10$ and $k = 50$. As observed, the state y_s achieves the predicted steady output

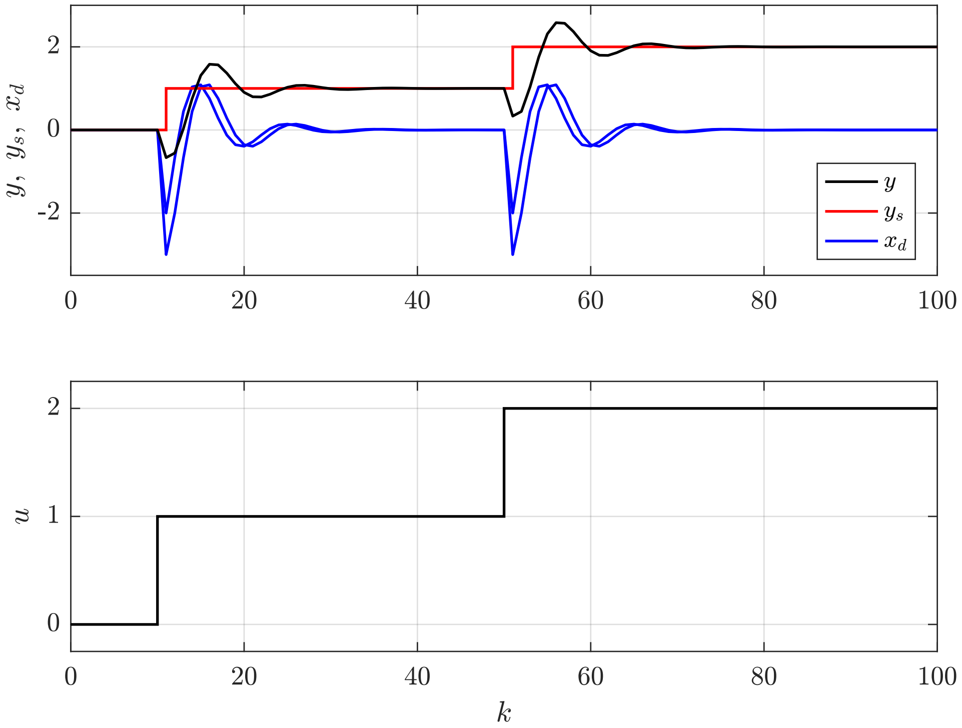

Figure 1 – System output, states and input of the new OPOM

that corresponds to the applied unit-step and x_d tends to zero because A is stable. Also, system output initially responds in the opposite direction due to the unstable zero, but eventually achieves the predicted steady output y_s, as expected.

2.3 The new OPOM for stable, integrating and unstable systems

Now, let us assume that the system (2.6)-(2.7) has n_{st} stable, n_{in} integrating and n_{un} unstable poles. There exists a matrix T such that the following similarity transformations can be written (GOLUB; LOAN, 2013, page 353):

$$T^{-1}x = \begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \\ \tilde{x}_{un} \end{bmatrix}, \quad T^{-1}AT = \begin{bmatrix} \tilde{A}_{st} & 0 & 0 \\ 0 & \tilde{A}_{in} & 0 \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix}, \quad T^{-1}B = \begin{bmatrix} \tilde{B}_{st} \\ \tilde{B}_{in} \\ \tilde{B}_{un} \end{bmatrix}$$

$$CT = \begin{bmatrix} \tilde{C}_{st} & \tilde{C}_{in} & \tilde{C}_{un} \end{bmatrix}$$

in which $\tilde{x}_{st} \in \mathbb{R}^{n_{st}}$, $\tilde{x}_{in} \in \mathbb{R}^{n_{in}}$, $\tilde{x}_{un} \in \mathbb{R}^{n_{un}}$, $\tilde{A}_{st} \in \mathbb{R}^{n_{st}\times n_{st}}$, $\tilde{A}_{in} \in \mathbb{R}^{n_{in}\times n_{in}}$, $\tilde{A}_{un} \in \mathbb{R}^{n_{un}\times n_{un}}$, $\tilde{B}_{st} \in \mathbb{R}^{n_{st}\times n_u}$, $\tilde{B}_{in} \in \mathbb{R}^{n_{in}\times n_u}$, $\tilde{B}_{un} \in \mathbb{R}^{n_{un}\times n_u}$, $\tilde{C}_{st} \in \mathbb{R}^{n_y\times n_{st}}$, $\tilde{C}_{in} \in \mathbb{R}^{n_y\times n_{in}}$ and $\tilde{C}_{un} \in \mathbb{R}^{n_y\times n_{un}}$. The transformation matrix T can be obtained from a real Schur decomposition of A followed by the block diagonalization using Roth's removal rule (GERRISH; WARD, 1998). More details can be found in Appendix A.

After performing the system decomposition, the following state-space model can be written:

$$\begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \\ \tilde{x}_{un} \end{bmatrix}^{+} = \begin{bmatrix} \tilde{A}_{st} & 0 & 0 \\ 0 & \tilde{A}_{in} & 0 \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} \begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \\ \tilde{x}_{un} \end{bmatrix} + \begin{bmatrix} \tilde{B}_{st} \\ \tilde{B}_{in} \\ \tilde{B}_{un} \end{bmatrix} u \tag{2.15}$$

$$y = \begin{bmatrix} \tilde{C}_{st} & \tilde{C}_{in} & \tilde{C}_{un} \end{bmatrix} \begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \\ \tilde{x}_{un} \end{bmatrix} \tag{2.16}$$

We shall mention that the tilde $(\tilde{\cdot})$ notation is used here to make a distinction between the notation we reserve for the final form of the OPOM model that will be obtained.

Let us define the incremental variables $\Delta\tilde{x}_{st}(k) = \tilde{x}_{st}(k) - \tilde{x}_{st}(k-1)$, $\Delta\tilde{x}_{in}(k) = \tilde{x}_{in}(k) - \tilde{x}_{in}(k-1)$ and $\Delta\tilde{x}_{un}(k) = \tilde{x}_{un}(k) - \tilde{x}_{un}(k-1)$. Again, considering a step input of amplitude Δu is applied to the system at time instant k and following the same steps as in the previous section, we have that:

$$\begin{aligned} y(k+j) = y(k) &+ \tilde{C}_{st}\left(I + \tilde{A}_{st} + \cdots + \tilde{A}_{st}^{j-2} + \tilde{A}_{st}^{j-1}\right)\left(\tilde{A}_{st}\Delta\tilde{x}_{st}(k) + \tilde{B}_{st}\Delta u(k)\right) \\ &+ \tilde{C}_{in}\left(I + \tilde{A}_{in} + \cdots + \tilde{A}_{in}^{j-2} + \tilde{A}_{in}^{j-1}\right)\left(\tilde{A}_{in}\Delta\tilde{x}_{in}(k) + \tilde{B}_{in}\Delta u(k)\right) \\ &+ \tilde{C}_{un}\left(I + \tilde{A}_{un} + \cdots + \tilde{A}_{un}^{j-2} + \tilde{A}_{un}^{j-1}\right)\left(\tilde{A}_{un}\Delta\tilde{x}_{un}(k) + \tilde{B}_{un}\Delta u(k)\right) \end{aligned} \tag{2.17}$$

Note that, for stable and unstable modes, we can substitute the partial sum given in (2.8), which does not apply for the integrating part since $\left(I - \tilde{A}_{in}\right)$ is singular. Thus, (2.17) can be rewritten as follows:

$$\begin{aligned} y(k+j) = y(k) &+ \tilde{C}_{st}\left(I - \tilde{A}_{st}^{j}\right)\left(I - \tilde{A}_{st}\right)^{-1}\left(\tilde{A}_{st}\Delta\tilde{x}_{st}(k) + \tilde{B}_{st}\Delta u(k)\right) \\ &+ \tilde{C}_{in}\sum_{i=0}^{j-1}\tilde{A}_{in}^{i}\left(\tilde{A}_{in}\Delta\tilde{x}_{in}(k) + \tilde{B}_{in}\Delta u(k)\right) \\ &+ \tilde{C}_{un}\left(I - \tilde{A}_{un}^{j}\right)\left(I - \tilde{A}_{un}\right)^{-1}\left(\tilde{A}_{un}\Delta\tilde{x}_{un}(k) + \tilde{B}_{un}\Delta u(k)\right) \end{aligned} \tag{2.18}$$

Now, let us define the following states:

$$y_s(k) = y(k) + \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{A}_{st}\Delta\tilde{x}_{st}(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{A}_{un}\Delta\tilde{x}_{un}(k) \tag{2.19}$$

$$x_{st}(k) = -\left(I - \tilde{A}_{st}\right)^{-1}\tilde{A}_{st}\Delta\tilde{x}_{st}(k) \tag{2.20}$$

$$x_{in}(k) = \Delta\tilde{x}_{in}(k) \tag{2.21}$$

$$x_{un}(k) = -\left(I - \tilde{A}_{un}\right)^{-1}\tilde{A}_{un}\Delta\tilde{x}_{un}(k) \tag{2.22}$$

Then, (2.18) can be rewritten as follows:

$$\begin{aligned}
y(k+j) = {} & y_s(k) + \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \\
& + \tilde{C}_{st}\tilde{A}_{st}^{j}\left(x_{st}(k) - \left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k)\right) \\
& + \tilde{C}_{in}\sum_{i=0}^{j-1}\tilde{A}_{in}^{i}\left(\tilde{A}_{in}x_{in}(k) + \tilde{B}_{in}\Delta u(k)\right) \\
& + \tilde{C}_{un}\tilde{A}_{un}^{j}\left(x_{un}(k) - \left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k)\right)
\end{aligned}$$

For $j = 0$, we have that:

$$\begin{aligned}
y(k) = {} & y_s(k) + \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \\
& + \tilde{C}_{st}\left(x_{st}(k) - \left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k)\right) \\
& + \tilde{C}_{un}\left(x_{un}(k) - \left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k)\right) \\
= {} & y_s(k) + \tilde{C}_{st}x_{st}(k) + \tilde{C}_{un}x_{un}(k)
\end{aligned}$$

In the same way, considering $j = 1$, it yields:

$$\begin{aligned}
y(k+1) = {} & y_s(k) + \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \\
& + \tilde{C}_{st}\tilde{A}_{st}\left(x_{st}(k) - \left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k)\right) \\
& + \tilde{C}_{in}\left(\tilde{A}_{in}x_{in}(k) + \tilde{B}_{in}\Delta u(k)\right) \\
& + \tilde{C}_{un}\tilde{A}_{un}\left(x_{un}(k) - \left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k)\right) \\
= {} & y_s(k+1) + \tilde{C}_{st}x_{st}(k+1) + \tilde{C}_{un}x_{un}(k+1)
\end{aligned}$$

with

$$\begin{aligned}
y_s(k+1) &= y_s(k) + \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \\
&\quad + \tilde{C}_{in}\left(\tilde{A}_{in}x_{in}(k) + \tilde{B}_{in}\Delta u(k)\right) \\
x_{st}(k+1) &= \tilde{A}_{st}x_{st}(k) - \tilde{A}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) \\
x_{un}(k+1) &= \tilde{A}_{un}x_{un}(k) - \tilde{A}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k)
\end{aligned}$$

Finally, we can write the following state-space model:

$$\begin{bmatrix} y_s \\ x_{st} \\ x_{in} \\ x_{un} \end{bmatrix}^{+} = \begin{bmatrix} I_{n_y} & 0 & C_{in}A_{in} & 0 \\ 0 & A_{st} & 0 & 0 \\ 0 & 0 & A_{in} & 0 \\ 0 & 0 & 0 & A_{un} \end{bmatrix} \begin{bmatrix} y_s \\ x_{st} \\ x_{in} \\ x_{un} \end{bmatrix} + \begin{bmatrix} B_s \\ B_{st} \\ B_{in} \\ B_{un} \end{bmatrix} \Delta u \tag{2.23}$$

$$y = \begin{bmatrix} I_{n_y} & C_{st} & 0 & C_{un} \end{bmatrix} \begin{bmatrix} y_s \\ x_{st} \\ x_{in} \\ x_{un} \end{bmatrix} \tag{2.24}$$

in which

$$\begin{gathered} A_{st} = \tilde{A}_{st},\ A_{in} = \tilde{A}_{in},\ A_{un} = \tilde{A}_{un}, \\ B_s = \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st} + \tilde{C}_{in}\tilde{B}_{in} + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}, \\ B_{st} = -\tilde{A}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st},\ B_{in} = \tilde{B}_{in},\ B_{un} = -\tilde{A}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}, \\ C_{st} = \tilde{C}_{st},\ C_{in} = \tilde{C}_{in},\ C_{un} = \tilde{C}_{un}. \end{gathered}$$

Here in this representation, y_s corresponds to the artificial integrating states related to the incremental form of inputs and x_{st}, x_{in} and x_{un} stand for the states related to stable, integrating and unstable modes of the original system, respectively.

This general version of OPOM can be used in many formulations of IHMPC with OPOM available in the literature, e.g., the ones proposed by Odloak (2004), González and Odloak (2009) and Carrapiço and Odloak (2005).

2.3.1 An alternative formulation

In González, Marchetti and Odloak (2007), the authors propose a different OPOM model in which the states x_{in} related to the integrating modes do not depend on any model parameters that describe the system. Although this method can be applied only when the system has simple integrating poles (i.e., when $\tilde{A}_{in}$ is the identity matrix), it covers most cases of integrating systems encountered in the process industry. Moreover, this approach is particularly useful to formulate a robust MPC for dealing with uncertainties in the integrating part of the system. Thus, in order to cover this case, we provide here an alternative formulation of the new OPOM that has the same feature as the model proposed in González, Marchetti and Odloak (2007).

First, observe that $\Delta\tilde{x}_{in}(k+1) = \Delta\tilde{x}_{in}(k) + \tilde{B}_{in}\Delta u(k)$ for the case of simple integrating poles and, from (2.15), we have that $\Delta\tilde{x}_{in}(k) = \tilde{B}_{in}u(k-1)$. Then, (2.18) can be rewritten as follows:

$$\begin{aligned} y(k+j) = y(k) &+ \tilde{C}_{st}\left(I - \tilde{A}_{st}^{j}\right)\left(I - \tilde{A}_{st}\right)^{-1}\left(\tilde{A}_{st}\Delta\tilde{x}_{st}(k) + \tilde{B}_{st}\Delta u(k)\right) \\ &+ j\tilde{C}_{in}\tilde{B}_{in}\Big(u(k-1) + \Delta u(k)\Big) \\ &+ \tilde{C}_{un}\left(I - \tilde{A}_{un}^{j}\right)\left(I - \tilde{A}_{un}\right)^{-1}\left(\tilde{A}_{un}\Delta\tilde{x}_{un}(k) + \tilde{B}_{un}\Delta u(k)\right) \end{aligned} \tag{2.25}$$

Now, consider the states y_s, x_{st} and x_{un} given in (2.19), (2.20) and (2.22), respectively and take $x_{in} = u(k-1)$. Then, we can rewrite (2.25) as follows:

$$\begin{aligned} y(k+j) = y_s(k) &+ \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \\ &+ \tilde{C}_{st}\tilde{A}_{st}^{j}\left(x_{st}(k) - \left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k)\right) \\ &+ j\tilde{C}_{in}\tilde{B}_{in}\Big(x_{in}(k) + \Delta u(k)\Big) \\ &+ \tilde{C}_{un}\tilde{A}_{un}^{j}\left(x_{un}(k) - \left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k)\right) \end{aligned}$$

As we did before, taking $j = 0$, it yields:

$$\begin{aligned} y(k) = y_s(k) &+ \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \\ &+ \tilde{C}_{st}\left(x_{st}(k) - \left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k)\right) \\ &+ \tilde{C}_{un}\left(x_{un}(k) - \left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k)\right) \\ = y_s(k) &+ \tilde{C}_{st}x_{st}(k) + \tilde{C}_{un}x_{un}(k) \end{aligned}$$

And, for $j = 1$, we have that:

$$\begin{aligned} y(k+1) = y_s(k) &+ \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \\ &+ \tilde{C}_{st}\tilde{A}_{st}\left(x_{st}(k) - \left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k)\right) \\ &+ \tilde{C}_{in}\tilde{B}_{in}\Big(x_{in}(k) + \Delta u(k)\Big) \\ &+ \tilde{C}_{un}\tilde{A}_{un}\left(x_{un}(k) - \left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k)\right) \\ = y_s(k+1) &+ \tilde{C}_{st}x_{st}(k+1) + \tilde{C}_{un}x_{un}(k+1) \end{aligned}$$

with

$$\begin{aligned} y_s(k+1) &= y_s(k) + \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \\ &\quad + \tilde{C}_{in}\tilde{B}_{in}\Big(x_{in}(k) + \Delta u(k)\Big) \\ x_{st}(k+1) &= \tilde{A}_{st}x_{st}(k) - \tilde{A}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}\Delta u(k) \\ x_{un}(k+1) &= \tilde{A}_{un}x_{un}(k) - \tilde{A}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}\Delta u(k) \end{aligned}$$

Thus, we can obtain the state-space model given as follows:

$$\begin{bmatrix} y_s \\ x_{st} \\ x_{in} \\ x_{un} \end{bmatrix}^+ = \begin{bmatrix} I_{n_y} & 0 & C_{in}B_{in} & 0 \\ 0 & A_{st} & 0 & 0 \\ 0 & 0 & I_{n_u} & 0 \\ 0 & 0 & 0 & A_{un} \end{bmatrix} \begin{bmatrix} y_s \\ x_{st} \\ x_{in} \\ x_{un} \end{bmatrix} + \begin{bmatrix} B_s \\ B_{st} \\ I_{n_u} \\ B_{un} \end{bmatrix} \Delta u \tag{2.26}$$

$$y = \begin{bmatrix} I_{n_y} & C_{st} & 0 & C_{un} \end{bmatrix} \begin{bmatrix} y_s \\ x_{st} \\ x_{in} \\ x_{un} \end{bmatrix} \tag{2.27}$$

in which

$$\begin{gathered} A_{st} = \tilde{A}_{st},\ A_{un} = \tilde{A}_{un}, \\ B_s = \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st} + \tilde{C}_{in}\tilde{B}_{in} + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}, \\ B_{st} = -\tilde{A}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st},\ B_{in} = \tilde{B}_{in},\ B_{un} = -\tilde{A}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}, \\ C_{st} = \tilde{C}_{st},\ C_{in} = \tilde{C}_{in},\ C_{un} = \tilde{C}_{un}. \end{gathered}$$

Note here that the dynamics of state x_{in} related to the integrating part do not depend on any matrix of the original system (2.15)-(2.16). Therefore, this model has the same applicability as the one proposed in González, Marchetti and Odloak (2007).

2.4 The new OPOM for time-delay systems

Consider here a time-delay system with the state updating equation given as follows:

$$\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}^{(0)}u(k) + \tilde{B}^{(1)}u(k-1) + \cdots + \tilde{B}^{(q)}u(k-q) \tag{2.28}$$

in which q is the largest time delay between any output-input pair of the system. Hence, assuming this system has stable, integrating and unstable poles, after performing a system decomposition (NAGAR; SINGH, 2004), the following model is obtained:

$$\begin{aligned} \begin{bmatrix} \tilde{x}_{st}(k+1) \\ \tilde{x}_{in}(k+1) \\ \tilde{x}_{un}(k+1) \end{bmatrix} &= \begin{bmatrix} \tilde{A}_{st} & 0 & 0 \\ 0 & \tilde{A}_{in} & 0 \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} \begin{bmatrix} \tilde{x}_{st}(k) \\ \tilde{x}_{in}(k) \\ \tilde{x}_{un}(k) \end{bmatrix} + \begin{bmatrix} \tilde{B}_{st}^{(0)} \\ \tilde{B}_{in}^{(0)} \\ \tilde{B}_{un}^{(0)} \end{bmatrix} u(k) \\ &+ \begin{bmatrix} \tilde{B}_{st}^{(1)} \\ \tilde{B}_{in}^{(1)} \\ \tilde{B}_{un}^{(1)} \end{bmatrix} u(k-1) + \cdots + \begin{bmatrix} \tilde{B}_{st}^{(q)} \\ \tilde{B}_{in}^{(q)} \\ \tilde{B}_{un}^{(q)} \end{bmatrix} u(k-q) \end{aligned} \tag{2.29}$$

$$y(k) = \begin{bmatrix} \tilde{C}_{st} & \tilde{C}_{in} & \tilde{C}_{un} \end{bmatrix} \begin{bmatrix} \tilde{x}_{st}(k) \\ \tilde{x}_{in}(k) \\ \tilde{x}_{un}(k) \end{bmatrix} \tag{2.30}$$

In order to build the model in the incremental form of inputs, we can use the same expressions of the general formulation of the new OPOM given in (2.23)-(2.24). Then, we obtain the following model:

$$x(k+1) = Ax(k) + B^{(0)}\Delta u(k) + B^{(1)}\Delta u(k-1) + \cdots + B^{(q)}\Delta u(k-q) \tag{2.31}$$

$$y(k) = Cx(k) \tag{2.32}$$

with

$$x(k) = \begin{bmatrix} y_s^T(k) & x_{st}^T(k) & x_{in}^T(k) & x_{un}^T(k) \end{bmatrix}^T,$$

$$A = \begin{bmatrix} I_{n_y} & 0 & C_{in}A_{in} & 0 \\ 0 & A_{st} & 0 & 0 \\ 0 & 0 & A_{in} & 0 \\ 0 & 0 & 0 & A_{un} \end{bmatrix}, \quad C = \begin{bmatrix} I_{n_y} & C_{st} & 0 & C_{un} \end{bmatrix},$$

$$B^{(i)} = \begin{bmatrix} B_s^{(i)} \\ B_{st}^{(i)} \\ B_{in}^{(i)} \\ B_{un}^{(i)} \end{bmatrix}, \quad i \in \mathbb{I}_{0:q}$$

in which

$$A_{st} = \tilde{A}_{st},\ A_{in} = \tilde{A}_{in},\ A_{un} = \tilde{A}_{un},$$
$$B_s^{(i)} = \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{B}_{st}^{(i)} + \tilde{C}_{in}\tilde{B}_{in}^{(i)} + \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{B}_{un}^{(i)},$$
$$B_{st}^{(i)} = -\left(I - \tilde{A}_{st}\right)^{-1}\tilde{A}_{st}\tilde{B}_{st}^{(i)},\ B_{in}^{(i)} = \tilde{B}_{in}^{(i)},\ B_{un}^{(i)} = -\left(I - \tilde{A}_{un}\right)^{-1}\tilde{A}_{un}\tilde{B}_{un}^{(i)},$$
$$C_{st} = \tilde{C}_{st},\ C_{in} = \tilde{C}_{in},\ C_{un} = \tilde{C}_{un}.$$

A conventional state-space representation can be obtained by augmenting the states in the same manner as presented in Santoro and Odloak (2012). For this purpose, the following additional states are defined:

$$\begin{bmatrix} z_1^T(k) & z_2^T(k) & \cdots & z_q^T(k) \end{bmatrix}^T = \begin{bmatrix} \Delta u^T(k-1) & \Delta u^T(k-2) & \cdots & \Delta u^T(k-q) \end{bmatrix}^T$$

Therefore, with a slight abuse of notation, the extended OPOM model for time-delay systems is given as follows:

$$x^+ = Ax + B\Delta u \tag{2.33}$$

$$y = Cx \tag{2.34}$$

with

$$x = \begin{bmatrix} y_s^T & x_{st}^T & x_{in}^T & x_{un}^T & z_1^T & z_2^T & \cdots & z_q^T \end{bmatrix}^T,$$

$$A = \begin{bmatrix} I_{n_y} & 0 & C_{in}A_{in} & 0 & B_s^{(1)} & B_s^{(2)} & \cdots & B_s^{(q-1)} & B_s^{(q)} \\ 0 & A_{st} & 0 & 0 & B_{st}^{(1)} & B_{st}^{(2)} & \cdots & B_{st}^{(q-1)} & B_{st}^{(q)} \\ 0 & 0 & A_{in} & 0 & B_{in}^{(1)} & B_{in}^{(2)} & \cdots & B_{in}^{(q-1)} & B_{in}^{(q)} \\ 0 & 0 & 0 & A_{un} & B_{un}^{(1)} & B_{un}^{(2)} & \cdots & B_{un}^{(q-1)} & B_{un}^{(q)} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{n_u} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & I_{n_u} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_s^{(0)} \\ B_{st}^{(0)} \\ B_{in}^{(0)} \\ B_{un}^{(0)} \\ I_{n_u} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix},$$

$$C = \begin{bmatrix} I_{n_y} & C_{st} & 0 & C_{un} & 0_{n_y \times n_u q} \end{bmatrix}$$

We shall note that, if the new OPOM for time-delay systems is built by using the alternative formulation given in (2.26)-(2.27), the resulting model is suitable for formulating a robust MPC for integrating systems with time delay, as proposed by Martins *et al.* (2013).

2.4.1 A comparison with the traditional OPOM

Here, we compare the new OPOM formulation with the traditional one proposed in Santoro and Odloak (2012) for integrating systems with time delay. For this purpose, consider the following system taken from Santoro and Odloak (2012):

$$y(s) = \frac{-0.19e^{-s}}{s\,(10s+1)}u_1(s) + \frac{0.235}{s\,(15s+1)}u_2(s) \tag{2.35}$$

Following the steps described in Santoro and Odloak (2012) and considering a sampling time $T_s = 1$, we obtain the traditional OPOM model given below:

$$x^+ = \begin{bmatrix} 1 & 0 & 0 & 1 & 1.71 & 0 \\ 0 & 0.9048 & 0 & 0 & -1.7192 & 0 \\ 0 & 0 & 0.9355 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -0.19 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & -3.29 \\ 0 & 0 \\ 0 & 3.2977 \\ 0 & 0.2350 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \Delta u \tag{2.36}$$

$$y = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} x \tag{2.37}$$

Now, we proceed by following the steps for building the new OPOM model given in (2.33)-(2.34). First, we consider $T_s = 1$ and discretize the transfer function model given in (2.35). Then, the following state-space realization is obtained in MATLAB®:

$$\tilde{x}(k+1) = \begin{bmatrix} 0.4132 & -0.4942 & 0.1610 \\ 0.8656 & 1.7905 & -0.2989 \\ 0.8656 & 0.8549 & 0.6366 \end{bmatrix} \tilde{x}(k) + \begin{bmatrix} 0 & -0.0212 \\ 0 & 0.0998 \\ 0 & -0.0252 \end{bmatrix} u(k)$$

$$+ \begin{bmatrix} -0.0568 & 0 \\ -0.0677 & 0 \\ -0.0677 & 0 \end{bmatrix} u(k-1)$$

$$y(k) = \begin{bmatrix} -0.0482 & 0.0887 & 0.0873 \end{bmatrix} \tilde{x}(k)$$

After performing a system decomposition into stable and integrating manifolds, we can write the model in the form given in (2.29)-(2.30) with the following system matrices:

$$\tilde{A}_{st} = \begin{bmatrix} 0.9048 & 0 \\ 0 & 0.9355 \end{bmatrix}, \ \tilde{A}_{in} = 1,$$

$$\tilde{B}_{st}^{(0)} = \begin{bmatrix} 0 & 0 \\ 0 & 1.7063 \end{bmatrix}, \ \tilde{B}_{in}^{(0)} = \begin{bmatrix} 0 & 1.7608 \end{bmatrix},$$

$$\tilde{B}_{st}^{(1)} = \begin{bmatrix} 1.3588 & 0 \\ 0 & 0 \end{bmatrix}, \ \tilde{B}_{in}^{(1)} = \begin{bmatrix} -1.4236 & 0 \end{bmatrix},$$

$$\tilde{C}_{st} = \begin{bmatrix} 0.1331 & -0.1332 \end{bmatrix}, \ \tilde{C}_{in} = 0.1335$$

The new OPOM model can be directly obtained by applying the formulas for computing the matrices used to build the model (2.33)-(2.34). This results in the following model:

$$x^{+} = \begin{bmatrix} 1 & 0 & 0 & 0.1335 & 1.71 & 0 \\ 0 & 0.9048 & 0 & 0 & -12.9197 & 0 \\ 0 & 0 & 0.9355 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1.4236 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & -3.29 \\ 0 & 0 \\ 0 & -24.7512 \\ 0 & 1.7608 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \Delta u \tag{2.38}$$

$$y = \begin{bmatrix} 1 & 0.1331 & -0.1332 & 0 & 0 & 0 \end{bmatrix} x \tag{2.39}$$

Figure 2 depicts the dynamics of the traditional OPOM (2.36)-(2.37) and the new OPOM (2.38)-(2.39) under the application of input increments $\Delta u = (-1, 0.1)$ and $\Delta u = (1, -0.1)$ at time instants $k = 10$ and $k = 20$, respectively. As expected, the outputs y of both models are identical, as well as the state y_s, which corresponds to the predicted steady output when the integrating state x_{in} is zeroed. Regarding the dynamics of states x_{st} and x_{in}, the difference between the models is due to the presence of larger absolute values in the state and input matrices of the new OPOM, which does not result in different output dynamics because the output matrix has proportionally smaller numbers.

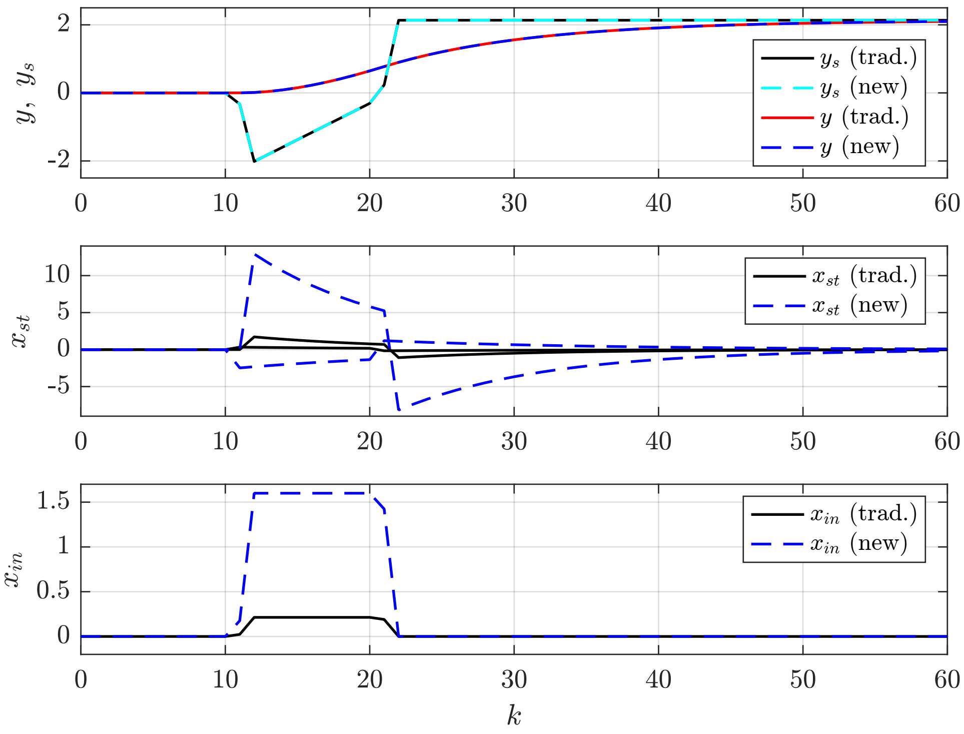

Figure 2 – System response comparison between the new OPOM and the traditional one

As observed in the above example, although the new OPOM and the traditional one produce the same input/output behavior, they may have different state/input gains. This depends on the initial state-space model that one chooses to build the new OPOM. Since state-space realizations are not unique, different models can be obtained through similarity transformations. For instance, note that one can obtain the same model matrices as the traditional OPOM by considering the following transformation matrix:

$$T = diag\left(\left[\tilde{C}_{st}\ \tilde{C}_{in}\right]\right) = \begin{bmatrix} 0.1331 & 0 & 0 \\ 0 & -0.1332 & 0 \\ 0 & 0 & 0.1335 \end{bmatrix}$$

and redefining the decomposed model as follows:

$$\begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \end{bmatrix} \leftarrow T \begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \end{bmatrix}, \quad \begin{bmatrix} \tilde{A}_{st} & 0 \\ 0 & \tilde{A}_{in} \end{bmatrix} \leftarrow T \begin{bmatrix} \tilde{A}_{st} & 0 \\ 0 & \tilde{A}_{in} \end{bmatrix} T^{-1},$$
$$\begin{bmatrix} \tilde{B}_{st} \\ \tilde{B}_{in} \end{bmatrix} \leftarrow T \begin{bmatrix} \tilde{B}_{st} \\ \tilde{B}_{in} \end{bmatrix}, \quad \begin{bmatrix} \tilde{C}_{st} & \tilde{C}_{in} \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{C}_{st} & \tilde{C}_{in} \end{bmatrix} T^{-1}$$

This results in the following transformed matrices:

$$\tilde{A}_{st} = \begin{bmatrix} 0.9048 & 0 \\ 0 & 0.9355 \end{bmatrix}, \ \tilde{A}_{in} = 1,$$
$$\tilde{B}_{st}^{(0)} = \begin{bmatrix} 0 & 0 \\ 0 & -0.2273 \end{bmatrix}, \ \tilde{B}_{in}^{(0)} = \begin{bmatrix} 0 & 0.2350 \end{bmatrix},$$

$$\tilde{B}_{st}^{(1)} = \begin{bmatrix} 0.1808 & 0 \\ 0 & 0 \end{bmatrix}, \; \tilde{B}_{in}^{(1)} = \begin{bmatrix} -0.19 & 0 \end{bmatrix},$$
$$\tilde{C}_{st} = \begin{bmatrix} 1 & 1 \end{bmatrix}, \; \tilde{C}_{in} = 1$$

Again, applying the formulas for building the new OPOM (2.33)-(2.34), we now obtain a model that is identical to the traditional OPOM given in (2.36)-(2.37).

2.5 Output feedback with the new OPOM

Similar to available MPC approaches that rely on a state-feedback framework, the deployment of the new OPOM into an MPC formulation also demands the state vector to be available. Hence, to apply these controllers in practice, one often needs to design state observers since only plant outputs are in general measured. However, it is well-known that closed-loop stability may not be assured by separately designing a stable state observer and a nominally stabilizing MPC. This is so because the MPC control law becomes nonlinear when there are active constraints, which makes the separation principle no longer applicable.

To avoid the need for designing state observers, YOUNG *et al.* (1987) proposed a non-minimal state-space model built in such a way that the state vector consists of only measured variables, namely, the current system output along with past outputs and inputs. This model is obtained as a state-space realization that is equivalent to the following discrete-time difference equation model:

$$y(k) = -\sum_{i=1}^{n_a} A_i y(k-i) + \sum_{i=1}^{n_b} B_i u(k-i) \tag{2.40}$$

in which $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$, $A_i \in \mathbb{R}^{n_y \times n_y}$ and $B_i \in \mathbb{R}^{n_y \times n_u}$ are matrices of model coefficients, and n_a and n_b are maximum delays with respect to outputs and inputs, respectively.

Let the state vector be defined as follows:

$$x(k) = \begin{bmatrix} y(k)^T & y(k-1)^T & \cdots & y(k-n_a)^T & u(k-1)^T & u(k-2)^T & \cdots & u(k-n_b)^T \end{bmatrix}^T$$

Then, model (2.40) can be written in the following state-space form:

$$x^+ = Ax + Bu \tag{2.41}$$
$$y = Cx \tag{2.42}$$

in which

$$A = \begin{bmatrix} -A_1 & -A_2 & \cdots & -A_{n_a-1} & -A_{n_a} & B_2 & \cdots & B_{n_b-1} & B_{n_b} \\ I_{n_y} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & I_{n_y} & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I_{n_y} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & I_{n_u} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & I_{n_u} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ I_{n_u} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$C = \begin{bmatrix} I_{n_y} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Since the state-space form (2.41)-(2.42) is obtained by realigning the model on past data, it is also known as the realigned model (MACIEJOWSKI, 2002). Although the state-space realization of model (2.40) presented here has inputs in the positional form, realigned models with incremental inputs have also been proposed (WANG; YOUNG, 2006; PEREZ; ODLOAK, 2006).

Finite horizon MPC formulations using the realigned model with embedded integral action for removing offset were proposed in Wang and Young (2006) and Zhang *et al.* (2011). Concerning the infinite horizon approach, an output-feedback robust MPC based on the realigned model was proposed in Perez and Odloak (2006) and González and Odloak (2010). The IHMPC with the realigned model is then extended to integrating systems for the nominal case (GONZÁLEZ; PEREZ; ODLOAK, 2009) and when multi-model uncertainty is considered (PEREZ; ODLOAK; LIMA, 2012; PEREZ; ODLOAK; LIMA, 2014). However, the formulations proposed by Perez, Odloak and Lima (2012) and Perez, Odloak and Lima (2014) consider systems that can be divided into pure integrating outputs and pure stable outputs, which poses a limitation on dealing with other types of dynamics. More recently, Ribeiro *et al.* (2020) employed the realigned model in the development of an output-feedback tube-based robust MPC.

Therefore, the use of the realigned model as a starting model to build the new OPOM represents an interesting alternative to deal with output feedback while also allowing the application of existing IHMPC formulations with OPOM, which cover a much wider class of systems and control objectives in comparison with available MPCs exclusively developed with realigned models. Note that the key property of the realigned model, i.e., the measured states, is not lost in this approach because, by construction, the states of the new OPOM can be written in terms of the states of the original system (2.6)-(2.7). To show that, consider equations (2.19)-(2.22), then observe that the states of the new OPOM can be computed as follows:

$$\begin{bmatrix} y_s \\ x_{st} \\ x_{in} \\ x_{un} \end{bmatrix} = L \begin{bmatrix} x \\ \Delta x \end{bmatrix} \tag{2.43}$$

with

$$L = \begin{bmatrix} I_{n_y} & \tilde{C}_{st}\left(I - \tilde{A}_{st}\right)^{-1}\tilde{A}_{st} & 0 & \tilde{C}_{un}\left(I - \tilde{A}_{un}\right)^{-1}\tilde{A}_{un} \\ 0 & -\left(I - \tilde{A}_{st}\right)^{-1}\tilde{A}_{st} & 0 & 0 \\ 0 & 0 & I_{n_{in}} & 0 \\ 0 & 0 & 0 & -\left(I - \tilde{A}_{un}\right)^{-1}\tilde{A}_{un} \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & T^{-1} \end{bmatrix}$$

in which T is the transformation matrix used to obtain system (2.15)-(2.16).

As an alternative, when A has inverse, the incremental states in (2.43) can be substituted by the predecessor input u^{-} as follows:

$$\begin{bmatrix} y_s \\ x_{st} \\ x_{in} \\ x_{un} \end{bmatrix} = \tilde{L} \begin{bmatrix} x \\ u^{-} \end{bmatrix} \tag{2.44}$$

with

$$\tilde{L} = L \begin{bmatrix} I_{n_y} & 0 \\ (I - A^{-1}) & A^{-1}B \end{bmatrix}$$

This shows that the state vector of the new OPOM can be directly computed whenever the states of the original system are measured. Note that this can be achieved not only with the realigned model but also when a state-space model is obtained through the linearization of a first-principle model written in terms of measured variables. On the other hand, when no state-space model with measured states is available, the new OPOM allows the use of an existing state observer originally designed for the (possibly minimal) original state-space model of the system. This means that, by using the new OPOM, one can avoid either the use of a state observer or the design of a new one, which represents an important advantage over the traditional OPOM, whose states must always be estimated via a tailor-designed observer.

2.6 Conclusion

We here proposed a new method to build a state-space model in the velocity form of inputs that has, as one of the states, the prediction of system output at steady-state. This approach successfully emulates the output prediction-oriented model, originally presented by Rodrigues and Odloak (2003b). The proposed model, here named as the new OPOM, has some advantages over the traditional OPOM proposed by Rodrigues and Odloak (2003b). They are summarized as follows:

- the new OPOM is built from a state-space model, which is a more general representation of MIMO systems and is usually used in the modern MPC literature;
- model construction is straightforward since one can easily plug system matrices into the final formulas;
- it is easy to automate in a computer program, which favors the development of an industrial package;
- this approach directly deals with pole multiplicity;
- if the original system is minimal, then the vector $\begin{bmatrix} x_{st}^T & x_{in}^T & x_{un}^T \end{bmatrix}^T$ is also minimal because the dimension of the original system is retained in the dynamic part, which may generate smaller models in comparison to the traditional OPOM;
- the new OPOM avoids the use of a state observer when the state of the original system is measured, making it possible to use the realigned model;
- it also supports the use of an existing state observer for the original system.

Under minimal adaptations, the new OPOM proposed in this chapter can be applied to existing IHMPC formulations based on the traditional OPOM. Moreover, the new OPOM allows the combination of such approaches into a more general IHMPC formulation. Since it is not in the scope of this chapter to present such an extended IHMPC formulation, the reader is referred to the works of Sencio and Odloak (2018a) and Sencio and Odloak (2018b). The application of the IHMPC with the new OPOM to a deisobutanizer column and to an unstable reactor is presented in Sencio and Odloak (2018a), while in Sencio and Odloak (2018b) the authors show the application to a system whose stable, integrating and unstable modes have double multiplicity. However, these approaches of IHMPC with the new OPOM are based on a finite control horizon, which means that integrating and unstable modes must be zeroed at the end of the prediction horizon so as to assure closed-loop stability. This not only produces a suboptimal controller but also results in a small domain of attraction. The latter issue can be alleviated by including slack variables into stability constraints, guaranteeing problem feasibility, which comes at the expense of losing stability guarantees for open-loop unstable systems. To address this problem, a more general dual-mode MPC with OPOM will be presented in the next chapter.

Chapter 3

A dual-mode MPC with OPOM

3.1 Introduction

In practical applications, an MPC controller typically receives steady-state targets from an optimizer that performs the integration between the MPC and an upper layer called real-time optimization (RTO) (YING; VOORAKARANAM; JOSEPH, 1999). Due to variations in operating conditions and economic objectives specified in the RTO stage, the desired targets may change and the controller should be able to steer the system to the new reference while satisfying constraints. Theoretically speaking, this objective is achieved by ensuring problem feasibility, closed-loop stability and offset removal.

Enjoying stability properties first established for the receding horizon regulator (RAWLINGS; MUSKE, 1993), early MPC formulations deal with the problem of target tracking simply by shifting system coordinates to the operating point provided by the target calculation layer (MUSKE; RAWLINGS, 1993; RAO; RAWLINGS, 1999). To obtain offset-free control in the presence of unmeasured disturbances and model uncertainty, this approach often requires augmenting the model with integrating disturbances that must be estimated to produce steady-state and input targets that are consistent with zero tracking error in the controlled variables. Criteria for designing disturbance models for offset-free control are given in Muske and Badgwell (2002) and Pannocchia and Rawlings (2003).

Although the offset in the controlled variables can be effectively removed by using disturbance models, the target optimizer *per se* does not ensure the feasibility of the control problem when large excursions between operating points are required, for instance, due to changing economic criteria in the RTO layer. In other words, during sudden changes in the reference, terminal stability constraints may not be satisfied, leading the control problem to lose its feasibility. This occurs because shifting the system origin to the new steady state also changes the controller's domain of attraction, which may turn the current state to be infeasible. In such cases, a longer control horizon must be chosen so that problem feasibility is recovered (SCOKAERT; RAWLINGS, 1998). Another issue with shifting the system origin is that the invariant set employed in a terminal inequality constraint must be recomputed, which may be intractable to be performed online.

Feasibility issues are handled in some approaches (ROSSITER; KOUVARITAKIS; GOSSNER, 1996; CHISCI; ZAPPA, 2003) by simply switching between different control laws according to whether the constraints can be satisfied or not for the current target and system state. If not, a feasibility recovery mechanism is triggered. Rossiter, Kouvaritakis and Gossner (1996) proposed an algorithm for SISO systems in which a constrained stable generalized predictive control (CSGPC) (ROSSITER; KOUVARITAKIS, 1993) is applied whenever the terminal constraint is feasible, otherwise, a modified endpoint constraint with a slack variable is employed. A contractive constraint guarantees the predicted steady output converges to the desired setpoint. Concerning target handling in MPC algorithms, Chisci and Zappa (2003) also propose a switching strategy in which, if the current state becomes infeasible upon a setpoint change, a recovery mode is activated to steer the state into the domain of attraction of a predictive regulator. Under some assumptions, it is proved that the algorithm ensures a finite recovery time.

To deal with the tracking problem in constrained systems, the so-called command governors ensure constraints satisfaction by computing admissible references to be tracked by the controller. Gilbert, Kolmanovsky and Tan (1995) proposed a first-order low-pass filter with a bandwidth parameter that depends nonlinearly on the system reference, which is modified to enforce state and input constraints. A generalized reference governor approach for nonlinear systems has been proposed by Gilbert and Kolmanovsky (2002) and several other formulations have appeared in the literature, see e.g. a comprehensive survey on reference and command governors (GARONE; CAIRANO; KOLMANOVSKY, 2017) and the references therein.

Some strategies merge the features of reference governors and MPC in a single optimization layer (RODRIGUES; ODLOAK, 2003b; ODLOAK, 2004; LIMON *et al.*, 2008; FERRAMOSCA *et al.*, 2009; MAYNE; FALUGI, 2016). In general, the idea behind these studies lies in the inclusion of artificial references as decision variables of the optimal control problem, consisting of an extra degree of freedom that allows the desired target to be tracked in an admissible way. In these approaches, state (or output) and input predicted trajectories are penalized as deviations from their respective artificial references, which are, in turn, penalized in terms of the deviation from the true steady-state target, possibly chosen by the RTO layer. Deviations from the true reference are treated as slack variables in some MPC approaches, such as the ones presented by Rodrigues and Odloak (2003b) and Odloak (2004) and in the feasibility recovery mode proposed by Rossiter, Kouvaritakis and Gossner (1996). Since slack variables correspond to extra degrees of freedom, the idea of artificial references is implicitly deployed in these studies.

Concerning the explicit use of artificial references within the MPC formulation, Limon *et al.* (2008) proposed an MPC for tracking in which artificial steady state and input are characterized through a suitable parameter, representing an additional decision variable of the control problem. The deviation between the artificial steady state and the desired one is penalized in the so-called offset cost function and the resulting control strategy steers the system to the closest admissible reference if the desired target is not admissible. Moreover, in this approach, terminal stability constraints do not depend on any exogenous reference, which allows the control problem to retain feasibility despite changes in the desired target. Stability is ensured by constraining the terminal state and the artificial references into an invariant set for tracking.

The effect of the offset cost function on the local optimality property of this class of MPC based on artificial references is studied in Ferramosca *et al.* (2009) and in Ferramosca *et al.* (2011), and an extension to the zone control problem is presented in Ferramosca *et al.* (2010). Simon, Lofberg and Glad (2014) propose an MPC for tracking in which the invariant set for tracking is replaced by a dynamically scaled terminal set. More recent works based on artificial references have been proposed considering robust stability (D'JORGE; FERRAMOSCA; GONZÁLEZ, 2017), stochastic (PAULSON; SANTOS; MESBAH, 2019) and nonlinear systems (LIMON *et al.*, 2018; KÖHLER; MÜLLER; ALLGÖWER, 2020).

Artificial references can be sensitive to uncertainties given that even small changes in the model parameters may lead to very different steady states and inputs (DUGHMAN; ROSSITER, 2015). In this case, since the MPC for tracking proposed by Limon *et al.* (2008) has no built-in integral action, this approach may compute artificial references that do not represent a steady state for the real plant, which leads to offset in the controlled variables, as in fact can be observed in the experimental results presented in Ferramosca *et al.* (2013). Therefore, similar to approaches that deploy a target optimization layer (MUSKE; RAWLINGS, 1993; RAO; RAWLINGS, 1999), the one-stage MPC based on artificial references of Limon *et al.* (2008) also needs an augmented model with disturbance estimates so that consistent artificial references can be computed, thereby removing the influence of model errors and unmeasured constant disturbances.

Aiming at providing offset-free control, the MPC proposed by Rodrigues and Odloak (2003b) is based on a state-space model in the velocity-form of inputs. Corresponding to a particular case of the general disturbance model (PANNOCCHIA, 2015), velocity models have the advantage of avoiding the computation of input references at steady state (GONZÁLEZ; ADAM; MARCHETTI, 2008). Rodrigues and Odloak (2003b) propose the output prediction-oriented model (OPOM), consisting of a different class of velocity models in which the integrating states that appear when writing the model in the incremental form of inputs correspond to the prediction of system outputs at steady state, as described in the previous chapter. This embedded output prediction at steady state avoids the use of an auxiliary steady-state model and is useful for imposing the terminal constraint in a stabilizing MPC formulation. Rodrigues and Odloak (2003b) propose to soften the terminal constraint with a slack variable, which is then penalized in the control cost function. This strategy enlarges the controller's domain of attraction by ensuring problem feasibility under any setpoint change. Initially proposed for open-loop stable systems, the MPC with OPOM has also been extended for integrating (CARRAPIÇO; ODLOAK, 2005), unstable (MARTINS; ODLOAK, 2016; SANTANA; MARTINS; ODLOAK, 2020a), and time-delay systems (GONZÁLEZ; ODLOAK, 2011; SANTORO; ODLOAK, 2012). Approaches for handling the zone control problem and optimizing input targets have been proposed (GONZÁLEZ; ODLOAK, 2009; MARTINS; ODLOAK, 2016) and robust stability has been addressed considering different classes of model uncertainty such as the multi-model paradigm (MARTINS; ZANIN; ODLOAK, 2014; MARTINS; ODLOAK, 2016) and the polytopic representation (RODRIGUES; ODLOAK, 2003b; ODLOAK, 2004; ALVAREZ; ODLOAK, 2010). These formulations of MPC with OPOM have proved to be implementable in practice as demonstrated in successful applications in laboratory-scale plants (MARTIN; ODLOAK; KASSAB, 2013; SILVA *et al.*, 2020) as well as in the oil refining industry (CARRAPIÇO *et al.*, 2009; MARTIN; ZANIN; ODLOAK, 2019).

Although it ensures the feasibility of terminal constraints, the main problem of including slack variables is that, when dealing with open-loop integrating and unstable systems, one may not guarantee that the control cost function is non-increasing, which is an important argument for stability proof. To overcome this issue, some works apply a two-step approach as a mechanism to enforce slack variables to decrease over time instants (CARRAPIÇO; ODLOAK, 2005; GONZÁLEZ; MARCHETTI; ODLOAK, 2007; SANTORO; ODLOAK, 2012). Considering integrating systems and assuming no input saturation, finite-time convergence of the slack variable related to integrating modes is proved in González, Marchetti and Odloak (2007) and Santoro and Odloak (2012). One-step strategies propose including a contractive constraint that forces slack variables to decrease with respect to the slack variable recomputed for the current state and shifted past input sequence (SANTORO; ODLOAK, 2012; MARTINS; ODLOAK, 2016). Although this procedure ensures the contractive constraint is always satisfied, the convergence of the slack variable is not guaranteed and strongly depends on the slack penalization in the cost function. Consequently, stability is only proved by assuming slack variables related to non-stable modes are zeroed.

Given the respective limitations of the MPC formulations proposed by Rodrigues and Odloak (2003b) and Limon *et al.* (2008), this chapter exploits the synergy between the two approaches by extending the MPC with OPOM proposed in Rodrigues and Odloak (2003b) with the idea of invariant set for tracking presented in Limon *et al.* (2008). While the former produces offset-free control and allows the characterization of artificial references that are independent of model parameters at steady state, the latter provides a method for computing an invariant set that does not depend on any exogenous reference. The resulting strategy is a dual-mode MPC for tracking with embedded integral action and guaranteed stability and feasibility under any setpoint change. In the proposed approach, the characterization of steady outputs and inputs is based only on terminal states and inputs, which is simpler than the one proposed by Limon *et al.* (2008) since equilibria parametrization is not necessary. Even in the presence of model errors and unmeasured disturbances, our method allows the computation of artificial references that are consistent with the true plant steady state, leading to a well-posed performance index, i.e., whose minimum corresponds to zero tracking error. Moreover, unlike most MPC formulations available in the literature, the proposed controller explicitly deals with constraints on input increments, which is especially important in applications of model predictive controllers in the process industry. We also provide an extension of the proposed MPC to deal with output control zones and optimizing input targets. It is proved that, if the desired operating point is not admissible, the proposed controller steers the system to the operating point that minimizes an offset cost function. Numerical examples are provided.

This chapter is organized as follows. The next section presents the system model and constraints along with the characterization of the artificial references, and definitions of the invariant set for tracking and performance index. In Section 3.3, the formulation of the dual-mode MPC with OPOM is presented followed by technical lemmas and theorem, with their respective proofs. In Section 3.4, the proposed controller is applied to a double-integrator system and a comparison with the MPC for tracking of Limon *et al.* (2008) is given. In Section 3.5, we address the control problem that deals with output zones and input targets. The extended formulation is presented along with theoretical analyses regarding recursive feasibility and closed-loop convergence. Section 3.6 shows the application of the extended controller to an unstable reactor system. Finally, conclusions are given in Section 3.7.

3.2 Preliminaries

3.2.1 System model

Consider a linear time-invariant system with n_{st} stable, n_{in} integrating and n_{un} unstable poles, n_y outputs, n_u inputs and a maximum input delay of q time steps. The OPOM model built for this system has the following form:

$$x^+ = Ax + B\Delta u \tag{3.1}$$

$$y = Cx \tag{3.2}$$

with

$$x = \begin{bmatrix} y_s^T & x_{st}^T & x_{in}^T & x_{un}^T & z_1^T & z_2^T & \cdots & z_q^T \end{bmatrix}^T,$$

$$A = \begin{bmatrix} I_{n_y} & 0 & C_{in}A_{in} & 0 & B_s^{(1)} & B_s^{(2)} & \cdots & B_s^{(q-1)} & B_s^{(q)} \\ 0 & A_{st} & 0 & 0 & B_{st}^{(1)} & B_{st}^{(2)} & \cdots & B_{st}^{(q-1)} & B_{st}^{(q)} \\ 0 & 0 & A_{in} & 0 & B_{in}^{(1)} & B_{in}^{(2)} & \cdots & B_{in}^{(q-1)} & B_{in}^{(q)} \\ 0 & 0 & 0 & A_{un} & B_{un}^{(1)} & B_{un}^{(2)} & \cdots & B_{un}^{(q-1)} & B_{un}^{(q)} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{n_u} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & I_{n_u} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_s^{(0)} \\ B_{st}^{(0)} \\ B_{in}^{(0)} \\ B_{un}^{(0)} \\ I_{n_u} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix},$$

$$C = \begin{bmatrix} I_{n_y} & C_{st} & 0 & C_{un} & 0_{n_y \times n_u q} \end{bmatrix}$$

in which $y_s \in \mathbb{R}^{n_y}$, $x_{st} \in \mathbb{R}^{n_{st}}$, $x_{in} \in \mathbb{R}^{n_{in}}$, $x_{un} \in \mathbb{R}^{n_{un}}$ and $z_i \in \mathbb{R}^{n_u}$ $\forall i \in \mathbb{I}_{1:q}$ are past input increments, with $z_i = \Delta u(k-i)$.

Here, we will not make any assumptions regarding the way the OPOM is obtained. In other words, this general form of OPOM can be built either by using the method proposed in the previous chapter (see Section 2.4) or by combining different formulations of the traditional OPOM proposed in the literature, namely the works of Odloak (2004), Santoro and Odloak (2012) and Martins and Odloak (2016).

To simplify our developments, we can group some matrices so that the system (3.1)-(3.2) is written as follows:

$$\begin{bmatrix} y_s \\ x_d \end{bmatrix}^+ = \begin{bmatrix} I_{n_y} & A_{sd} \\ 0 & A_d \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} + \begin{bmatrix} B_s \\ B_d \end{bmatrix} \Delta u \tag{3.3}$$

$$y = \begin{bmatrix} I_{n_y} & C_d \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} \tag{3.4}$$

in which $x_d \in \mathbb{R}^{n_d}$, $A_d \in \mathbb{R}^{n_d \times n_d}$, $A_{sd} \in \mathbb{R}^{n_y \times n_d}$, $B_s \in \mathbb{R}^{n_y \times n_u}$, $B_d \in \mathbb{R}^{n_d \times n_u}$, $C_d \in \mathbb{R}^{n_y \times n_d}$ and $n_d = n_{st} + n_{in} + n_{un} + n_u q$.

Assumption 1. The pair (A_d, B_d) is stabilizable.

3.2.2 System constraints

We consider the system is subject to linear constraints on system outputs, inputs and input increments, for which the following sets can be defined:

$$\mathcal{Y} := \{y \in \mathbb{R}^{n_y} : A_y y \leq b_y\} \tag{3.5}$$

$$\mathcal{U} := \{u \in \mathbb{R}^{n_u} : A_u u \leq b_u\} \tag{3.6}$$

$$\mathcal{U}_\Delta := \{\Delta u \in \mathbb{R}^{n_u} : A_\Delta \Delta u \leq b_\Delta\} \tag{3.7}$$

Note that, unlike most MPC formulations available in the literature, constraints on input increments are explicitly considered here. This type of constraint is associated with physical limitations and equipment preservation, and is mostly used by practitioners in the process industry. Among the reasons for constraining input increments are: control valves may have a limited positioning velocity; the motor coupled to the pump shaft has limited acceleration; furnaces and reboilers must have moderate temperature variations in order to avoid equipment damage due to thermal dilatation of tubes and refractory walls; etc.

Remark 2. Due to the inevitable presence of disturbances and model errors in practical applications, hard output constraints may lead to infeasibilities of the control problem, which is the reason why this type of constraint usually is not taken into account in MPC formulations with OPOM. Since the approach proposed here does not depend on the existence of output constraints, they can either be removed or softened. Different strategies for handling infeasibilities of MPC problems are discussed in Scokaert and Rawlings (1999).

3.2.3 Characterization of steady outputs and inputs

To simplify notation, for a given state $x = (y_s, x_d)$, let u^- denote the predecessor input, i.e. the input of previous time step with respect to x. Here, we will show that steady outputs and inputs can be computed as $(\bar{y}_s, \bar{u}_s) = M(x, u^-)$, in which M is a suitable matrix that will be defined later on.

First, consider the following control law:

$$\Delta u = K x_d \tag{3.8}$$

in which K is computed such that $F = A_d + B_d K$ is discrete-time Hurwitz. Note that Assumption 1 guarantees the existence of a stabilizing control gain K, whose computation can be performed by means of standard techniques such as pole placement or the well-known LQR (linear quadratic regulator). Under control law (3.8), the evolution of system (3.3) is described by the following autonomous system:

$$x^+ = \begin{bmatrix} I_{n_y} & A_{sd} + B_s K \\ 0 & F \end{bmatrix} x$$

For a given $x = x(k) = (y_s(k), x_d(k))$, states $y_s(k+j)$ and $x_d(k+j)$ are given separately as follows:

$$y_s(k+j) = y_s(k) + (A_{sd} + B_s K) \sum_{i=0}^{j-1} F^i x_d(k) \tag{3.9}$$

$$x_d(k+j) = F^j x_d(k) \tag{3.10}$$

Noting that $\sum_{i=0}^{j-1} F^i = (I - F^j)(I - F)^{-1} = (I - F)^{-1}(I - F^j)$ and defining $\Gamma = (I - F)^{-1}$, we can compute the output $y(k+j)$ as follows:

$$\begin{aligned} y(k+j) &= y_s(k+j) + C_d x_d(k+j) \\ &= y_s(k) + \Big((A_{sd} + B_s K)\Gamma + \left(C_d - (A_{sd} + B_s K)\Gamma\right) F^j \Big) x_d(k) \end{aligned} \tag{3.11}$$

Since F is a stable matrix, we can compute the steady output as follows:

$$\bar{y}_s = \lim_{j\to\infty} y_s(k+j) = y_s(k) + (A_{sd} + B_s K)\Gamma x_d(k) \tag{3.12}$$

In the same manner, for a given $u^- = u(k-1)$, the input $u(k+j)$ is:

$$\begin{aligned} u(k+j) = u(k-1) + \sum_{i=0}^{j} \Delta u(k+i) &= u(k-1) + K \sum_{i=0}^{j} F^i x_d(k) \\ &= u(k-1) + K\Gamma(I - F^{j+1}) x_d(k) \end{aligned} \tag{3.13}$$

And the steady input can be computed as follows:

$$\bar{u}_s = \lim_{j\to\infty} u(k+j) = u(k-1) + K\Gamma x_d(k) \tag{3.14}$$

Then, the combination of (3.12) and (3.14) results in:

$$(\bar{y}_s, \bar{u}_s) = M(x, u^-) \tag{3.15}$$

with

$$M = \begin{bmatrix} I_{n_y} & (A_{sd} + B_s K)\Gamma & 0_{n_y \times n_u} \\ 0_{n_u \times n_y} & K\Gamma & I_{n_u} \end{bmatrix}$$

Note that, when the system is at steady state, $x_d = 0$ and $\bar{y}_s$ and $\bar{u}_s$ do not depend on any model parameter, becoming the actual steady output and input of the system. Thus, even in the presence of plant-model mismatch, unbiased steady output and input are obtained.

3.2.4 Characterization of the invariant set for tracking

For the characterization of the invariant set for tracking, we will follow similar steps as the approach proposed by Limon *et al.* (2008). However, we here use a different characterization of steady-state variables and explicitly consider constraints on input increments.

The evolution of states and inputs considering system (3.3) and control law (3.8) is given in terms of the augmented state $w = (x, u^-)$ as follows:

$$w^+ = A_w w \tag{3.16}$$

in which

$$A_w = \begin{bmatrix} I_{n_y} & A_{sd} + B_s K & 0 \\ 0 & F & 0 \\ 0 & K & I_{n_u} \end{bmatrix}$$

Now, considering the sets given in (3.5)-(3.7), the following convex polyhedron can be defined for any $\lambda \in (0, 1)$:

$$\mathcal{W}_\lambda := \Big\{ w = (x, u^-) : Cx \in \mathcal{Y}, (u^- + Kx_d) \in \mathcal{U}, Kx_d \in \mathcal{U}_\Delta, Mw \in \lambda(\mathcal{Y} \times \mathcal{U}) \Big\} \quad (3.17)$$

A set $\Omega_t^w \subseteq \mathcal{W}_{\lambda=1}$ is said to be admissible positively invariant for system (3.16) if, for all $w \in \Omega_t^w$, the successor $w^+ = A_w w$ is such that $w^+ \in \Omega_t^w$. The maximal admissible invariant set for tracking is defined as $\mathcal{O}_\infty^w = \{w : A_w^i w \in \mathcal{W}_{\lambda=1}, \forall i \geq 0\}$.

Maximal admissible invariant sets can be computed offline by applying the algorithm proposed in Gilbert and Tan (1991). However, observing the sufficient conditions for finite determinability of maximal admissible invariant sets provided in Gilbert and Tan (1991), the set $\mathcal{O}_\infty^w$ may not be finitely determined because A_w has $n_y + n_u$ eigenvalues on the unit circle that are related to y_s and u^-. To deal with such cases, Gilbert and Tan (1991) propose a method to compute an approximation of the maximal admissible invariant set and, by using this approach, we can ensure that the set $\mathcal{O}_{\infty,\lambda}^w := \{w : A_w^i w \in \mathcal{W}_\lambda, \ \forall i \geq 0\}$ can be finitely determined for any $\lambda \in (0, 1)$. In other words, by choosing λ arbitrarily close to 1, one can obtain a polyhedral approximation to the maximal invariant set $\mathcal{O}_\infty^w$, which has also been applied in the computation of the invariant set for tracking proposed in Limon *et al.* (2008).

Given a $\Omega_t^w = \mathcal{O}_{\infty,\lambda}^w$ for some $\lambda \in (0, 1)$, the set of steady outputs that can be tracked with no steady error is limited to $\bar{\mathcal{Y}}_s = \text{Proj}_y(\bar{\mathcal{Z}}_s)$, in which $\bar{\mathcal{Z}}_s$ is the set of admissible steady outputs and inputs $(\bar{y}_s, \bar{u}_s)$ that are consistent with the invariant set for tracking and is defined as follows:

$$\bar{\mathcal{Z}}_s := \Big\{ \bar{z}_s = (\bar{y}_s, \bar{u}_s) : (\bar{y}_s, \bar{u}_s) = M(x, u^-), \ (x, u^-) \in \Omega_t^w, \ u^- \in \mathcal{U} \Big\} \quad (3.18)$$

3.2.5 Performance index

Let N denote a control horizon along which we can define the following sequence of input increments:

$$\Delta\mathbf{u} = \{\Delta u(0), \Delta u(1), \ldots, \Delta u(N-1)\} \quad (3.19)$$

Given a state x, a sequence of input movements $\Delta\mathbf{u}$ and an output setpoint y_{sp}, let us consider the following infinite horizon performance index:

$$V(x, \Delta\mathbf{u}, y_{sp}) = \sum_{j=0}^{\infty} \|y(j) - y_{sp}\|_{Q_y}^2 + \|\Delta u(j)\|_R^2 \quad (3.20)$$

in which $Q_y \in \mathbb{R}^{n_y \times n_y}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are given weighting matrices and the outputs $y(j) \in \mathbb{R}^{n_y}$ are computed through model (3.3)-(3.4) with $x(0) = x$ and control movements $\Delta u(j)$ that are elements of the sequence $\Delta\mathbf{u}$ for $j \in \mathbb{I}_{0:N-1}$ and are given as $\Delta u(j) = Kx_d(j)$ for $j \in \mathbb{I}_{\geq N}$.

In order to obtain a finite horizon performance index, we can split (3.20) into partial and infinite sums:

$$\begin{aligned} V_N(x, \Delta\mathbf{u}, y_{sp}) = & \sum_{j=0}^{N-1} \|y(j) - y_{sp}\|_{Q_y}^2 + \|\Delta u(j)\|_R^2 \\ & + \sum_{j=0}^{\infty} \|y(N+j) - y_{sp}\|_{Q_y}^2 + \|\Delta u(N+j)\|_R^2 \end{aligned} \tag{3.21}$$

Combining (3.11) with (3.12) and noting that $\Delta u(N+j) = F^j x_d(N)$, the infinite sum of (3.21) can be written as follows:

$$\begin{aligned} & \sum_{j=0}^{\infty} \|y(N+j) - y_{sp}\|_{Q_y}^2 + \|\Delta u(N+j)\|_R^2 = \\ & \qquad \sum_{j=0}^{\infty} \|\bar{y}_s + (C_d - (A_{sd} + B_s K)\Gamma)\, F^j x_d(N+j) - y_{sp}\|_{Q_y}^2 \\ & \qquad + \|KF^j x_d(N+j)\|_R^2 \end{aligned} \tag{3.22}$$

To prevent the above infinite sum from being unbounded, we must ensure that $\bar{y}_s - y_{sp} = 0$. Thus, the infinite sum can be computed as follows:

$$\sum_{j=0}^{\infty} \|y(N+j) - y_{sp}\|_{Q_y}^2 + \|\Delta u(N+j)\|_R^2 = \|x_d(N)\|_P^2 \tag{3.23}$$

in which

$$P = \sum_{j=0}^{\infty} (F^j)^T \bar{Q} F^j \tag{3.24}$$

with

$$\bar{Q} = (C_d - (A_{sd} + B_s K)\Gamma)^T Q_y (C_d - (A_{sd} + B_s K)\Gamma) + K^T R K \tag{3.25}$$

Since F is stable, the infinite sum P converges and can thus be computed. For this purpose, multiply P by F^T from the left and by F from the right and observe that:

$$P - F^T P F = \sum_{j=0}^{\infty} (F^j)^T \bar{Q} F^j - \sum_{j=1}^{\infty} (F^j)^T \bar{Q} F^j$$

which results in the following Lyapunov equation:

$$P - F^T P F = \bar{Q} \tag{3.26}$$

Then, the performance index (3.21) can be written as follows:

$$V_N(x, \Delta\mathbf{u}, y_{sp}) = \sum_{j=0}^{N-1} \left(\|y(j) - y_{sp}\|_{Q_y}^2 + \|\Delta u(j)\|_R^2 \right) + \|x_d(N)\|_P^2 \tag{3.27}$$

Note that $\bar{y}_s - y_{sp} = 0$ is a terminal constraint because $\bar{y}_s$ is defined in terms of the terminal state $x(N) = (y_s(N), x_d(N))$. Due to conflicts with input constraints (3.6) and (3.7), this terminal constraint may not be satisfied in a control problem, leading to infeasibilities. To circumvent this issue, this terminal constraint can be relaxed as $\bar{y}_s - y_{sp} - \delta = 0$, in which $\delta \in \mathbb{R}^{n_y}$ is an additional degree of freedom that corresponds to the offset between the steady output $\bar{y}_s$ and the output setpoint y_{sp}. A similar type of relaxation was first proposed by Rodrigues and Odloak (2003b) for stable systems without time delay in which no terminal control law is employed and the terminal constraint is simply given as $y_s(N) - y_{sp} - \delta = 0$. From (3.12), it is easy to see that this terminal constraint proposed in Rodrigues and Odloak (2003b) is a particular case of the one we employed in this work since $\bar{y}_s = y_s(N)$ for $A_{sd} = 0$ and $K = 0$.

As in Rodrigues and Odloak (2003b), the slack variable δ is penalized by a matrix $S_y \in \mathbb{R}^{n_y \times n_y}$ in the performance index, which can be redefined as follows:

$$V_N(x, \Delta\mathbf{u}, y_{sp}, \delta) = \sum_{j=0}^{N-1} \left(\|y(j) - y_{sp} - \delta\|^2_{Q_y} + \|\Delta u(j)\|^2_R \right) + \|x_d(N)\|^2_P + \|\delta\|^2_{S_y} \tag{3.28}$$

Although the inclusion of the slack variable δ serves the purpose of explicitly showing the relaxation of the terminal constraint, it can be eliminated from the performance index by substituting $\delta = \bar{y}_s - y_{sp}$. Hence, the performance index can be written as follows:

$$V_N(x, \Delta\mathbf{u}, y_{sp}) = \sum_{j=0}^{N-1} \left(\|y(j) - \bar{y}_s\|^2_{Q_y} + \|\Delta u(j)\|^2_R \right) + \|x_d(N)\|^2_P + \|\bar{y}_s - y_{sp}\|^2_{S_y} \tag{3.29}$$

Note that the steady output $\bar{y}_s$ can be viewed here as an artificial reference, which is a much more meaningful quantity than δ. Also, observe that the last term of (3.29), which corresponds to the penalization of the slack variable δ employed in Rodrigues and Odloak (2003b), is equivalent to the offset cost function proposed in Limon *et al.* (2008). This shows the similarity between the two approaches.

In some applications, one may want to penalize the input trajectory rather than input increments. Then, for the sake of completeness, the performance index can be extended as follows:

$$\begin{aligned} V_N(w, \Delta\mathbf{u}, y_{sp}) = &\sum_{j=0}^{N-1} \left(\|y(j) - \bar{y}_s\|^2_{Q_y} + \|u(j) - \bar{u}_s\|^2_{Q_u} + \|\Delta u(j)\|^2_R \right) \\ &+ \|x_d(N)\|^2_P + \|\bar{y}_s - y_{sp}\|^2_{S_y} \end{aligned} \tag{3.30}$$

in which $Q_u \in \mathbb{R}^{n_u \times n_u}$ and P is computed as solution of the following Lyapunov equation:

$$P - F^T P F = \bar{Q} \tag{3.31}$$

where

$$\bar{Q} = (C_d - (A_{sd} + B_s K)\Gamma)^T Q_y (C_d - (A_{sd} + B_s K)\Gamma) + (K\Gamma F)^T Q_u (K\Gamma F) + K^T R K \tag{3.32}$$

3.3 A dual-mode MPC with OPOM

3.3.1 Problem formulation

For a given extended state $w = (x, u^-)$ and an output setpoint y_{sp}, the proposed dual-mode MPC with OPOM is based on the control problem $P_N(w, y_{sp})$ defined as follows:

$$\Delta\mathbf{u}^* = \arg\min_{\Delta\mathbf{u}} \quad V_N(w, \Delta\mathbf{u}, y_{sp}) \tag{3.33}$$

subject to

$$\begin{aligned}
&(x(0), u(-1)) = w && && (3.34)\\
&x(j+1) = Ax(j) + B\Delta u(j), && j \in \mathbb{I}_{0:N-1} && (3.35)\\
&y(j) = Cx(j), && j \in \mathbb{I}_{0:N-1} && (3.36)\\
&u(j) = u(j-1) + \Delta u(j), && j \in \mathbb{I}_{0:N-1} && (3.37)\\
&y(j) \in \mathcal{Y}, && j \in \mathbb{I}_{0:N-1} && (3.38)\\
&u(j) \in \mathcal{U}, && j \in \mathbb{I}_{0:N-1} && (3.39)\\
&\Delta u(j) \in \mathcal{U}_\Delta, && j \in \mathbb{I}_{0:N-1} && (3.40)\\
&(\bar{y}_s, \bar{u}_s) = M(x(N), u(N-1)) && && (3.41)\\
&(x(N), u(N-1)) \in \Omega_t^w && && (3.42)
\end{aligned}$$

in which $\Omega_t^w = \mathcal{O}_{\infty,\lambda}^w$ for some $\lambda \in (0, 1)$.

The set of initial extended states w that can be admissibly steered in N steps to Ω_t^w defines a polyhedral region $\mathcal{W}_N \subseteq \mathbb{R}^{n_y+n_d+n_u}$ such that problem $P_N(w, y_{sp})$ is feasible for all $w \in \mathcal{W}_N$. Note that y_{sp} does not affect problem feasibility since no constraint depends on y_{sp}. Then, for a given extended state $w_k \in \mathcal{W}_N$ and output setpoint $y_{sp,k}$ at time step k, the solution of problem $P_N(w_k, y_{sp,k})$ results in an optimal sequence of control movements $\Delta\mathbf{u}_k^*$. Since only the first element $\Delta u^*(0|k)$ is applied to the system and this procedure is repeated in a receding horizon fashion, it implicitly defines the control law $\Delta u(k) := \kappa_N(w_k, y_{sp,k})$.

Remark 3. Note that feedforward control can be incorporated into problem $P_N(w, y_{sp})$ by considering a measured disturbance as a fixed input that remains constant over predictions.

3.3.2 Recursive feasibility and convergence analyses

Before establishing the recursive feasibility of problem $P_N(w, y_{sp})$ and the system convergence under the control law $\kappa_N(w, y_{sp})$, we will make the following assumption:

Assumption 2. Let the following hold:

1. System (3.3)-(3.4) is undisturbed and the state is measured;
2. $K \in \mathbb{R}^{n_u \times n_d}$ is a stabilizing control gain for the pair (A_d, B_d);
3. $Q_y \in \mathbb{R}^{n_y \times n_y}$, $P \in \mathbb{R}^{n_d \times n_d}$, $S_y \in \mathbb{R}^{n_y \times n_y}$ are positive definite matrices with P computed through the Lyapunov equation (3.31);
4. Matrices $Q_u \in \mathbb{R}^{n_u \times n_u}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are given such that both are positive definite or one of them is positive definite and the other is a positive semidefinite matrix.

Now, the following technical lemmas and theorem can be stated:

Lemma 1 (Recursive feasibility)**.** *Let Assumption 2 hold and consider a time step k with an extended state $w_k \in \mathcal{W}_N$ and an output setpoint $y_{sp,k}$. Then, the feasibility of problem $P_N(w_k, y_{sp,k})$ at time step k implies that $P_N(w_{k+j}, y_{sp,k+j})$ will remain feasible at any subsequent time step $k+j$, $\forall j \in \mathbb{I}_{\geq 1}$.*

Proof. This proof follows standard arguments found in the MPC literature. Let $\Delta\mathbf{u}_k^*$ denote the solution of problem $P_N(w_k, y_{sp,k})$ at time step k. By the receding horizon principle, only $\Delta u^*(0|k)$ is injected into the system and we move to time step $k+1$, at which $x(0|k+1) = x(k+1) = x(1|k)$. Consider a suboptimal candidate solution to problem $P_N(w_{k+1}, y_{sp,k+1})$ denoted as $\Delta\tilde{\mathbf{u}}_{k+1}$ and given as follows:

$$\Delta\tilde{\mathbf{u}}_{k+1} = \{\Delta u^*(1|k), \ldots, \Delta u^*(N-1|k), Kx_d(N|k)\} \tag{3.43}$$

Since we have explicitly defined $\Delta\tilde{u}(j|k+1) = \Delta u^*(j+1|k)$ for $j \in \mathbb{I}_{0:N-2}$ in (3.43), it is easy to see that $u(j|k+1) = u(j+1|k) \in \mathcal{U}\ \forall j \in \mathbb{I}_{0:N-2}$ and, because the system is assumed to be undisturbed, $x(j|k+1) = x(j+1|k)\ \forall j \in \mathbb{I}_{0:N-1}$ and $y(j|k+1) = y(j+1|k) \in \mathcal{Y}\ \forall j \in \mathbb{I}_{0:N-1}$. Concerning the last element of $\Delta\tilde{\mathbf{u}}_{k+1}$, observe that $(x(N|k), u(N-1|k)) \in \Omega_t^w$ and Ω_t^w is an admissible invariant set that satisfies (3.17), which implies that $\Delta\tilde{u}(N-1|k+1) = Kx_d(N|k) \in \mathcal{U}_\Delta$ and $u(N-1|k+1) = (u(N-1|k) + Kx_d(N|k)) \in \mathcal{U}$. Consequently, constraints (3.34)-(3.40) are satisfied. Since (3.41) only defines transformation of variables, it is trivially satisfied as well. Finally, due to the positive invariance of Ω_t^w, we have that $(x(N|k+1),\ u(N-1|k+1)) = (x(N+1|k),\ u(N|k)) \in \Omega_t^w$ and, thus, constraint (3.42) is also satisfied.

Therefore, by induction, it is easy to see that problem $P_N(w_{k+j}, y_{sp,k+j})$ will remain feasible at every time step $k+j$, $\forall j \in \mathbb{I}_{\geq 1}$. □

Lemma 2 (Auxiliary results). *Let Assumption 2 hold and consider a given extended state $w = (x, u^-) \in int(\Omega_t^w)$ in which $x = (\bar{y}_s, 0)$ and an admissible output setpoint $y_{sp} \in \bar{\mathcal{Y}}_s$ with $\bar{y}_s \neq y_{sp}$. Then, there exists an admissible sequence of control movements given as $\Delta\tilde{\mathbf{u}} = \{\Delta u, Kx_d^+, \dots, KF^{N-2}x_d^+\}$ that produces a new artificial reference $\bar{y}_s^+$ such that the following relationship holds:*

$$\|\bar{y}_s - \bar{y}_s^+\|_{Q_y}^2 + \|u - \bar{u}_s^+\|_{Q_u}^2 + \|\Delta u\|_R^2 + \|x_d^+\|_P^2 + \|\bar{y}_s^+ - y_{sp}\|_{S_y}^2 < \|\bar{y}_s - y_{sp}\|_{S_y}^2 \tag{3.44}$$

in which $(\bar{y}_s^+, \bar{u}_s^+) = M(x^+, u)$, with $x^+ = (y_s^+, x_d^+)$, $y_s^+ = \bar{y}_s + B_s\Delta u$, $x_d^+ = B_d\Delta u$ and $u = u^- + \Delta u$.

Proof. First, we will prove that $\Delta\tilde{\mathbf{u}}$ is an admissible control sequence. For this purpose, consider a steady output given as $\bar{y}_s^+ = \alpha\bar{y}_s + (1-\alpha)y_{sp}$, in which $\alpha \in (0,1)$. We also have $\bar{y}_s^+ = \bar{y}_s + (B_s + (A_{sd} + B_sK)\,\Gamma B_d)\,\Delta u$ from the substitution of $x^+ = (y_s^+, x_d^+)$, $y_s^+ = \bar{y}_s + B_s\Delta u$ and $x_d^+ = B_d\Delta u$ into $\bar{y}_s^+ = M_y x^+$ where $M_y = \begin{bmatrix} I_{n_y} & (A_{sd} + B_sK)\,\Gamma \end{bmatrix}$. Noting that $\bar{y}_s^+ - \bar{y}_s = (1-\alpha)(y_{sp} - \bar{y}_s)$ and assuming $(B_s + (A_{sd} + B_sK)\,\Gamma B_d)$ has full rank, then there exists Δu such that $(1-\alpha)(y_{sp} - \bar{y}_s) = (B_s + (A_{sd} + B_sK)\,\Gamma B_d)\,\Delta u$ holds for some $\alpha \in (0,1)$.

Now, to prove that Δu is admissible, observe that, since $(x, u^-) \in \text{int}(\Omega_t^w)$, there exists a constant $\gamma \in (0,1)$ such that $(x, u^-) \in \gamma\Omega_t^w$. Then, taking a sufficiently large $\alpha \in (0,1)$ such that $(B\Delta u, \Delta u) \in (1-\gamma)\Omega_t^w$ and $\Delta u \in \mathcal{U}_\Delta$, it follows that $(x, u^-) + (B\Delta u, \Delta u) = (x^+, u) \in \Omega_t^w$, which implies that Δu is feasible. Now, since the remaining control movements of $\Delta\tilde{\mathbf{u}}$ are defined through the control law $\Delta u(j) = KF^{j-1}x_d^+$, $\forall j \in \mathbb{I}_{1:N-1}$, they are also feasible due to the positive invariance of Ω_t^w. Therefore, the entire sequence of control movements $\Delta\tilde{\mathbf{u}}$ is admissible, which proves the claim.

Using (3.10)-(3.14), the performance index computed for $\Delta\tilde{\mathbf{u}}$ is given as follows:

$$\begin{aligned} V_N(w, \Delta\tilde{\mathbf{u}}, y_{sp}) = & \sum_{j=0}^{N-1} \left(\|y(j) - \bar{y}_s^+\|_{Q_y}^2 + \|u(j) - \bar{u}_s^+\|_{Q_u}^2 + \|\Delta u(j)\|_R^2 \right) \\ & + \|x_d(N)\|_P^2 + \|\bar{y}_s^+ - y_{sp}\|_{S_y}^2 \\ = & \|\bar{y}_s - \bar{y}_s^+\|_{Q_y}^2 + \|(C_d - (A_{sd} + B_sK)\Gamma)\,x_d^+\|_{Q_y}^2 \\ & + \|(C_d - (A_{sd} + B_sK)\Gamma)\,Fx_d^+\|_{Q_y}^2 + \dots \\ & + \|(C_d - (A_{sd} + B_sK)\Gamma)\,F^{N-2}x_d^+\|_{Q_y}^2 + \|u - \bar{u}_s^+\|_{Q_u}^2 + \|K\Gamma Fx_d^+\|_{Q_u}^2 \\ & + \|K\Gamma F^2x_d^+\|_{Q_u}^2 + \dots + \|K\Gamma F^{N-1}x_d^+\|_{Q_u}^2 + \|\Delta u\|_R^2 + \|Kx_d^+\|_R^2 \\ & + \|KFx_d^+\|_R^2 + \dots + \|KF^{N-2}x_d^+\|_R^2 + \|F^{N-1}x_d^+\|_P^2 + \|\bar{y}_s^+ - y_{sp}\|_{S_y}^2 \end{aligned} \tag{3.45}$$

Now, considering (3.24) with $\bar{Q}$ given in (3.32), it follows that:

$$\begin{aligned}
\|(C_d - (A_{sd} + B_sK)\Gamma)\, x_d^+\|_{Q_y}^2 &+ \|(C_d - (A_{sd} + B_sK)\Gamma)\, Fx_d^+\|_{Q_y}^2 + \dots \\
&+ \|(C_d - (A_{sd} + B_sK)\Gamma)\, F^{N-2}x_d^+\|_{Q_y}^2 + \|u - \bar{u}_s^+\|_{Q_u}^2 \\
&+ \|K\Gamma F x_d^+\|_{Q_u}^2 + \|K\Gamma F^2 x_d^+\|_{Q_u}^2 + \dots + \|K\Gamma F^{N-1}x_d^+\|_{Q_u}^2 \\
&+ \|Kx_d^+\|_R^2 + \|KFx_d^+\|_R^2 + \dots + \|KF^{N-2}x_d^+\|_R^2 \\
&+ \|F^{N-1}x_d^+\|_P^2 \\
&= \left(x_d^+\right)^T \Bigg(\bar{Q} + F^T\bar{Q}F + \dots + \left(F^{N-2}\right)^T \bar{Q}F^{N-2} \\
&+ \sum_{j=N-1}^{\infty} \left(F^j\right)^T \bar{Q}F^j\Bigg) x_d^+ \\
&= \left(x_d^+\right)^T \left(\sum_{j=0}^{\infty} \left(F^j\right)^T \bar{Q}F^j\right) x_d^+ = \|x_d^+\|_P^2
\end{aligned}$$

Thus, the performance index given in (3.45) can be written as follows:

$$V_N(w, \Delta\tilde{\mathbf{u}}, y_{sp}) = \|\bar{y}_s - \bar{y}_s^+\|_{Q_y}^2 + \|u - \bar{u}_s^+\|_{Q_u}^2 + \|\Delta u\|_R^2 + \|x_d^+\|_P^2 + \|\bar{y}_s^+ - y_{sp}\|_{S_y}^2 \quad (3.46)$$

Noting that $\bar{y}_s^+ = M_y(y_s^+, x_d^+)$, we have:

$$\begin{aligned}
\|\bar{y}_s - \bar{y}_s^+\|_{Q_y}^2 &= \|\bar{y}_s - y_s^+ - (A_{sd} + B_sK)\Gamma x_d^+\|_{Q_y}^2 \\
&= \|\bar{y}_s - \bar{y}_s - B_s\Delta u - (A_{sd} + B_sK)\Gamma B_d\Delta u\|_{Q_y}^2 \\
&= \|\Delta u\|_{\tilde{Q}_y}^2 \qquad (3.47)
\end{aligned}$$

with $\tilde{Q}_y = (B_s + (A_{sd} + B_sK)\Gamma B_d)^T\, Q_y\, (B_s + (A_{sd} + B_sK)\Gamma B_d)$.

In the same manner, given that $\bar{u}_s^+ = M_u(x_d^+, u)$, with $M_u = \begin{bmatrix} K\Gamma & I_{n_u} \end{bmatrix}$, and $x_d^+ = B_d\Delta u$, we also have:

$$\begin{aligned}
\|u - \bar{u}_s^+\|_{Q_u}^2 &= \|u - K\Gamma x_d^+ - u\|_{Q_u}^2 \\
&= \|K\Gamma B_d\Delta u\|_{Q_u}^2 \\
&= \|\Delta u\|_{\tilde{Q}_u}^2 \qquad (3.48)
\end{aligned}$$

with $\tilde{Q}_u = (K\Gamma B_d)^T\, Q_u\, (K\Gamma B_d)$.

Using $x_d^+ = B_d\Delta u$ again, it follows that:

$$\|x_d^+\|_P^2 = \|B_d\Delta u\|_P^2 = \|\Delta u\|_{\tilde{P}}^2 \quad (3.49)$$

in which $\tilde{P} = B_d^TPB_d$.

Moreover, since $\bar{y}_s^+ - y_{sp} = \alpha(\bar{y}_s - y_{sp})$, the following expression can be obtained:

$$\begin{aligned}
\|\bar{y}_s^+ - y_{sp}\|_{S_y}^2 &= \alpha^2\|\bar{y}_s - y_{sp}\|_{S_y}^2 \\
&= \alpha^2(1-\alpha)^{-2}\|(B_s + (A_{sd} + B_sK)\Gamma B_d)\,\Delta u\|_{S_y}^2 \\
&= \alpha^2(1-\alpha)^{-2}\|\Delta u\|_{\tilde{S}_y}^2 \qquad (3.50)
\end{aligned}$$

with $\tilde{S}_y = (B_s + (A_{sd} + B_s K)\Gamma B_d)^T \, S_y \, (B_s + (A_{sd} + B_s K)\Gamma B_d)$.

The substitution of (3.47)-(3.50) into (3.46) yields:

$$\begin{aligned} V_N(w, \Delta\tilde{\mathbf{u}}, y_{sp}) &= \|\Delta u\|^2_{\tilde{Q}_y} + \|\Delta u\|^2_{\tilde{Q}_u} + \|\Delta u\|^2_R + \|\Delta u\|^2_{\tilde{P}} + \alpha^2(1-\alpha)^{-2}\|\Delta u\|^2_{\tilde{S}_y} \\ &= \|\Delta u\|^2_H \end{aligned} \tag{3.51}$$

in which $H = \tilde{Q}_y + \tilde{Q}_u + R + \tilde{P} + \alpha^2(1-\alpha)^{-2}\tilde{S}_y$.

Now, to prove that $\|\Delta u\|^2_H < \|\bar{y}_s - y_{sp}\|^2_{S_y} = (1-\alpha)^{-2}\|\Delta u\|^2_{\tilde{S}_y}$, we will show that there exists a sufficiently large $\alpha \in (0,1)$ such that $(1-\alpha)^{-2}\tilde{S}_y > H$. For this purpose, observe that there exists a constant $\varphi > 0$ such that $\varphi\tilde{S}_y > \tilde{Q}_y + \tilde{Q}_u + R + \tilde{P}$ holds for a positive definite matrix $\tilde{S}_y$. Thus, the following expression can be written:

$$\begin{aligned} (1-\alpha)^{-2}\tilde{S}_y - H &= (1-\alpha^2)(1-\alpha)^{-2}\tilde{S}_y - \tilde{Q}_y - \tilde{Q}_u - R - \tilde{S}_y \\ &> (1-\alpha^2)(1-\alpha)^{-2}\tilde{S}_y - \varphi\tilde{S}_y \\ &= (1-\alpha)^{-2}\left(1 - \alpha^2 - (1-\alpha)^2\varphi\right)\tilde{S}_y \\ &= (1-\alpha)^{-1}\left(1 + \alpha - (1-\alpha)\,\varphi\right)\tilde{S}_y \end{aligned}$$

Therefore, $(1-\alpha)^{-2}\tilde{S}_y > H$ for every $\alpha \in (\alpha_{min}, 1)$ with $\alpha_{min} = \max\left(0, \frac{\varphi-1}{\varphi+1}\right)$, which finishes the proof. □

Lemma 3 (Convergence of $\bar{y}_s$ to y_{sp})**.** *Under Assumption 2, consider a given initial extended state $w = (x, u^-) \in int(\Omega^w_t)$ with $x = (y_s, 0)$ and an admissible output setpoint $y_{sp} \in \bar{\mathcal{Y}}_s$. If the solution of problem $P_N(w, y_{sp})$ is $\Delta\mathbf{u}^*$ such that $\|Cx - \bar{y}_s\|^2_{Q_y} = 0$, then $\|\bar{y}_s - y_{sp}\|^2_{S_y} = 0$.*

Proof. Note that $x = (y_s, 0)$ is a steady state with corresponding steady output given by y_s. Since $\|Cx - \bar{y}_s\|^2_{Q_y} = 0$, it follows that $y_s = \bar{y}_s$, which implies that $\bar{u}_s = u^-$ and $\Delta\mathbf{u}^* = \{0, \ldots, 0\}$. To prove by contradiction, we will assume that $\bar{y}_s \neq y_{sp}$, which results in the performance index $V^*_N(w, \Delta\mathbf{u}^*, y_{sp}) = \|\bar{y}_s - y_{sp}\|^2_{S_y}$.

Since $w \in \text{int}(\Omega^w_t)$ and $\bar{y}_s \neq y_{sp}$, by virtue of Lemma 2, there exists an admissible sequence of control movements $\Delta\tilde{\mathbf{u}}$ that is suboptimal and produces an artificial reference $\bar{y}^+_s$ with the corresponding performance index $V_N(w, \Delta\tilde{\mathbf{u}}, y_{sp})$ given as follows:

$$\begin{aligned} V^*_N(w, \Delta\mathbf{u}^*, y_{sp}) &\le V_N(w, \Delta\tilde{\mathbf{u}}, y_{sp}) \\ &= \sum_{j=0}^{N-1}\left(\|y(j) - \bar{y}^+_s\|^2_{Q_y} + \|u(j) - \bar{u}^+_s\|^2_{Q_u} + \|\Delta u(j)\|^2_R\right) \\ &\quad + \|x_d(N)\|^2_P + \|\bar{y}^+_s - y_{sp}\|^2_{S_y} \\ &= \|\bar{y}_s - \bar{y}^+_s\|^2_{Q_y} + \|u - \bar{u}^+_s\|^2_{Q_u} + \|\Delta u\|^2_R + \|x^+_d\|^2_P + \|\bar{y}^+_s - y_{sp}\|^2_{S_y} \end{aligned}$$

in which $(\bar{y}^+_s, \bar{u}^+_s) = M(x^+, u)$, with $x^+ = (y^+_s, x^+_d)$, $y^+_s = \bar{y}_s + B_s\Delta u$, $x^+_d = B_d\Delta u$ and $u = u^- + \Delta u$.

Observe that, again by virtue of Lemma 2, the following relationship holds:

$$\begin{aligned}V_N^*(w,\Delta\mathbf{u}^*,y_{sp}) &\leq \|\bar{y}_s-\bar{y}_s^+\|_{Q_y}^2+\|u-\bar{u}_s^+\|_{Q_u}^2+\|\Delta u\|_R^2+\|x_d^+\|_P^2+\|\bar{y}_s^+-y_{sp}\|_{S_y}^2\\ &< \|\bar{y}_s-y_{sp}\|_{S_y}^2\\ &= V_N^*(w,\Delta\mathbf{u}^*,y_{sp})\end{aligned}$$

Since the strict inequality contradicts the optimality of $V_N^*(w,\Delta\mathbf{u}^*,y_{sp})$, it follows that $\bar{y}_s=y_{sp}$. □

Theorem 1 (Closed-loop convergence)**.** *Let Assumption 2 hold and consider an admissible output setpoint $y_{sp}\in\bar{\mathcal{Y}}_s$. Then, for any feasible initial augmented state $w_0\in\mathcal{W}_N$, the system* (3.3)-(3.4) *with inputs computed through control law $\kappa_N(w,y_{sp})$ is admissibly steered to the setpoint y_{sp}.*

Proof. Consider a given initial augmented state $w_k\in\mathcal{W}_N$ at time step k at which problem $P_N(w_k,y_{sp})$ is feasible with optimal solution denoted as $\Delta\mathbf{u}_k^*$ and optimal value function $V_N^*(w_k,\Delta\mathbf{u}_k^*,y_{sp})$. Then, by the receding horizon principle, only the first input movement is injected into the system and we move to time step $k+1$, at which a solution to problem $P_N(w_{k+1},y_{sp})$ is guaranteed to exist by virtue of Lemma 1.

Consider a suboptimal solution $\Delta\tilde{\mathbf{u}}_{k+1}$ to problem $P_N(w_{k+1},y_{sp})$ at time step $k+1$, in which $\Delta\tilde{\mathbf{u}}_{k+1}$ is given in (3.43). As already shown in Lemma 1, the sequence of control increments $\Delta\tilde{\mathbf{u}}_{k+1}$ is feasible. Let $\tilde{V}_N(w_{k+1},\Delta\tilde{\mathbf{u}}_{k+1},y_{sp})$ denote the corresponding performance criterion.

Now, we will show that $\Delta\tilde{\mathbf{u}}_{k+1}$ produces steady outputs and inputs such that $(\bar{y}_{s,k+1},\bar{u}_{s,k+1})=(\bar{y}_{s,k},\bar{u}_{s,k})$. Consider the matrix $M_y=\begin{bmatrix}I_{n_y} & (A_{sd}+B_sK)\,\Gamma\end{bmatrix}$. Since $\bar{y}_{s,k}=M_y(y_s(N|k),x_d(N|k))$ and $\bar{y}_{s,k+1}=M_y(y_s(N|k+1),x_d(N|k+1))$, with $y_s(N|k+1)=y_s(N|k)+(A_{sd}+B_sK)x_d(N|k)$ and $x_d(N|k+1)=Fx_d(N|k)$, the subtraction $\bar{y}_{s,k+1}-\bar{y}_{s,k}$ can be given as follows:

$$\bar{y}_{s,k+1}-\bar{y}_{s,k}=(A_{sd}+B_sK)\,(I+\Gamma(F-I))\,x_d(N|k)$$

Then, since $I+\Gamma(F-I)=I+(I-F)^{-1}(F-I)=I-(I-F)^{-1}(I-F)=0$, it follows that $\bar{y}_{s,k+1}=\bar{y}_{s,k}$. By following the same rationale, it can be shown that $\bar{u}_{s,k+1}=\bar{u}_{s,k}$.

Observe that $\Delta\tilde{\mathbf{u}}_{k+1}$ produces the following sequences of predicted outputs and inputs:

$$\begin{aligned}\{y(0|k+1),y(1|k+1),\ldots,y(N-1|k+1)\}&=\{y(1|k),y(2|k),\ldots,y(N|k)\}\\ \{u(0|k+1),u(1|k+1),\ldots,u(N-1|k+1)\}&=\{u(1|k),u(2|k),\ldots,u(N|k)\}\end{aligned}$$

Also, given that $x_d(N|k+1)=x_d(N+1|k)=Fx_d(N|k)$, the performance index $\tilde{V}_N(w_{k+1},\Delta\tilde{\mathbf{u}}_{k+1},y_{sp})$ can be written as follows:

$$\begin{aligned}\tilde{V}_N(w_{k+1},\Delta\tilde{\mathbf{u}}_{k+1},y_{sp}) = &\left\|y(1|k)-\bar{y}_{s,k}\right\|_{Q_y}^2+\cdots+\left\|y(N|k)-\bar{y}_{s,k}\right\|_{Q_y}^2\\ &+\left\|u(1|k)-\bar{u}_{s,k}\right\|_{Q_u}^2+\cdots+\left\|u(N|k)-\bar{u}_{s,k}\right\|_{Q_u}^2\\ &+\left\|\Delta u^*(1|k)\right\|_R^2+\cdots+\left\|\Delta u^*(N-1|k)\right\|_R^2+\left\|Kx_d(N|k)\right\|_R^2\\ &+\left\|Fx_d(N|k)\right\|_P^2+\left\|\bar{y}_{s,k}-y_{sp}\right\|_{S_y}^2 \qquad (3.52)\end{aligned}$$

Note that $\|y(N|k)-\bar{y}_{s,k}\|_{Q_y}^2$ can be given as follows:

$$\begin{aligned}\left\|y(N|k)-\bar{y}_{s,k}\right\|_{Q_y}^2 &= \left\|y_s(N|k)+C_dx_d(N|k)-y_s(N|k)-(A_{sd}+B_sK)\,\Gamma x_d(N|k)\right\|_{Q_y}^2\\ &= \left\|x_d(N|k)\right\|_{(C_d-(A_{sd}+B_sK)\Gamma)^TQ_y(C_d-(A_{sd}+B_sK)\Gamma)}^2 \qquad (3.53)\end{aligned}$$

Similarly, $\|u(N|k)-\bar{u}_{s,k}\|_{Q_u}^2$ can be written as follows:

$$\begin{aligned}\left\|u(N|k)-\bar{u}_{s,k}\right\|_{Q_u}^2 &= \left\|u_s(N-1|k)+K\Gamma\,(I-F)\,x_d(N|k)-u_s(N-1|k)-K\Gamma x_d(N|k)\right\|_{Q_u}^2\\ &= \left\|x_d(N|k)\right\|_{(K\Gamma F)^TQ_u(K\Gamma F)}^2 \qquad (3.54)\end{aligned}$$

Also, we have that:

$$\left\|Kx_d(N|k)\right\|_R^2 = \left\|x_d(N|k)\right\|_{K^TRK}^2 \qquad (3.55)$$

Summing up (3.53), (3.54) and (3.55) yields:

$$\left\|y(N|k)-\bar{y}_{s,k}\right\|_{Q_y}^2+\left\|u(N|k)-\bar{u}_{s,k}\right\|_{Q_u}^2+\left\|Kx_d(N|k)\right\|_R^2 = \left\|x_d(N|k)\right\|_{\bar{Q}}^2 \qquad (3.56)$$

with $\bar{Q}$ given in (3.32).

Combining (3.52) and (3.56) and noting that, from (3.31), $\bar{Q}+F^TPF=P$, it follows that:

$$\begin{aligned}\tilde{V}_N(w_{k+1},\Delta\tilde{\mathbf{u}}_{k+1},y_{sp}) = &\left\|y(1|k)-\bar{y}_{s,k}\right\|_{Q_y}^2+\cdots+\left\|y(N-1|k)-\bar{y}_{s,k}\right\|_{Q_y}^2\\ &+\left\|u(1|k)-\bar{u}_{s,k}\right\|_{Q_u}^2+\cdots+\left\|u(N-1|k)-\bar{u}_{s,k}\right\|_{Q_u}^2\\ &+\left\|\Delta u^*(1|k)\right\|_R^2+\cdots+\left\|\Delta u^*(N-1|k)\right\|_R^2\\ &+\left\|x_d(N|k)\right\|_{\bar{Q}}^2+\left\|Fx_d(N|k)\right\|_P^2+\left\|\bar{y}_{s,k}-y_{sp}\right\|_{S_y}^2\\ = &\sum_{j=1}^{N-1}\left(\left\|y(j|k)-\bar{y}_{s,k}\right\|_{Q_y}^2+\left\|u(j|k)-\bar{u}_{s,k}\right\|_{Q_u}^2+\left\|\Delta u^*(j|k)\right\|_R^2\right)\\ &+\left\|x_d(N|k)\right\|_P^2+\left\|\bar{y}_{s,k}-y_{sp}\right\|_{S_y}^2 \qquad (3.57)\end{aligned}$$

Now, we can easily show that the following relationship holds:

$$\begin{aligned}\tilde{V}_N(w_{k+1},\Delta\tilde{\mathbf{u}}_{k+1},y_{sp})-V_N^*(w_k,\Delta\mathbf{u}_k^*,y_{sp}) = &-\left\|y(0|k)-\bar{y}_{s,k}\right\|_{Q_y}^2-\left\|u(0|k)-\bar{u}_{s,k}\right\|_{Q_u}^2\\ &-\left\|\Delta u^*(0|k)\right\|_R^2 \qquad (3.58)\end{aligned}$$

Then, it follows that $\tilde{V}_N(w_{k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, y_{sp}) \leq V_N^*(w_k, \Delta\mathbf{u}_k^*, y_{sp})$ since the right-hand side of (3.58) is non-positive. By the optimality of $V_N^*(w_{k+1}, \Delta\mathbf{u}_{k+1}^*, y_{sp})$, we also have that $V_N^*(w_{k+1}, \Delta\mathbf{u}_{k+1}^*, y_{sp}) \leq \tilde{V}_N(w_{k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, y_{sp})$, which implies that:

$$\begin{aligned} V_N^*(w_{k+1}, \Delta\mathbf{u}_{k+1}^*, y_{sp}) - V_N^*(w_k, \Delta\mathbf{u}_k^*, y_{sp}) \leq &- \|y(0|k) - \bar{y}_{s,k}\|_{Q_y}^2 - \|u(0|k) - \bar{u}_{s,k}\|_{Q_u}^2 \\ &- \|\Delta u^*(0|k)\|_R^2 \end{aligned} \tag{3.59}$$

The expression in (3.59) implies that the sequence of performance criteria at subsequent time instants is non-increasing, which means that:

$$V_N^*(w_{k+1}, \Delta\mathbf{u}_{k+1}^*, y_{sp}) \leq V_N^*(w_k, \Delta\mathbf{u}_k^*, y_{sp}), \quad \forall k \in \mathbb{N}$$

Since the sequence of performance indexes is non-increasing and bounded below by zero, it converges, i.e. $\lim_{k\to\infty}\left(V_N^*(w_{k+1}, \Delta\mathbf{u}_{k+1}^*, y_{sp}) - V_N^*(w_k, \Delta\mathbf{u}_k^*, y_{sp})\right) = 0$, which implies that both sides of (3.59) tend to zero as $k \to \infty$. Consequently, by the positiveness of Q_y, it follows that $\lim_{k\to\infty}\|y(k) - \bar{y}_{s,k}\| = 0$. Moreover, if Q_u is positive definite, $\lim_{k\to\infty}\|u(k) - \bar{u}_{s,k}\| = 0$, which implies that $\lim_{k\to\infty}\|\Delta u(k)\| = 0$. On the other hand, if R is positive definite, $\lim_{k\to\infty}\|\Delta u(k)\| = 0$, implying that $\lim_{k\to\infty}\|x_d(k)\| = 0$. Consequently, from (3.15), we also have that $\lim_{k\to\infty}\|u(k) - \bar{u}_{s,k}\| = 0$.

Now, in virtue of Lemma 3, it follows that $\lim_{k\to\infty}\|y(k) - y_{sp}\| = 0$, which proves that the system output converges to the setpoint y_{sp}. □

3.4 Case study 1: double-integrator system

Here, we first compare the proposed controller (C1) with the MPC for tracking (C2) presented in Limon *et al.* (2008). For this purpose, consider the following double-integrator system:

$$\begin{aligned} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0.5 \\ 1 & 0.5 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned}$$

This system is subject to constraints on outputs and inputs, with $\|y\|_\infty \leq 5$ and $\|u\|_\infty \leq 0.3$. In the application of the proposed MPC, the tuning parameters have been chosen as $N = 4$, $Q_y = I_2$, $Q_u = I_2$, $R = 0 \times I_2$, and $S_y = 100 \times I_2$ and an LQR controller with $Q_{LQR} = R_{LQR} = I_2$ has been used to design the control law (3.8). A $\lambda = 0.99$ has been considered in the computation of the invariant set for tracking Ω_t^w. Using the notation of Limon *et al.* (2008), the controller C2 has been tuned with $N = 3$, $Q = I_2$, $R = I_2$, $T = 100 \times I_2$ and $\lambda = 0.99$. The terminal control law has also been designed using an

LQR controller with $Q_{LQR} = R_{LQR} = I_2$. These design parameters are the same as those used by Limon *et al.* (2008). System states are assumed to be measured and C1 uses the OPOM states computed through (2.43).

First, to compare C1 and C2, we will reproduce here the simulation scenario presented in Limon *et al.* (2008). The system starts from $x = (0, -2)$ and the output setpoint is initially $y_{sp} = (5, 0)$, which is changed to $y_{sp} = (-5.5, 0)$ at time step $k = 30$ and to $y_{sp} = (2, 0)$ at $k = 60$. In C1, we consider an initial input position of $u^- = (0.25, 0.1)$, which has no effect on the results since input increments are not constrained here for a fair comparison with C2.

Output and input responses obtained with controllers C1 and C2 are depicted in Figures 3 and 4, respectively. Observe that C1 and C2 produced the same results, steering the system to the closest (in a least square sense) admissible steady output such that $\bar{y}_s \in \bar{\mathcal{Y}}_s \subset \lambda\mathcal{Y}$. These steady outputs are $\bar{y}_s = (4.95, 0)$ when the setpoint is $y_{sp} = (5, 0)$ and $\bar{y}_s = (-4.95, 0)$ for $y_{sp} = (-5.5, 0)$. When the reference was changed to $y_{sp} = (2, 0) \in \bar{\mathcal{Y}}_s$, the system was driven to y_{sp} without steady error. As shown in Figure 5, the performance indexes are non-increasing for a given y_{sp}, as expected.

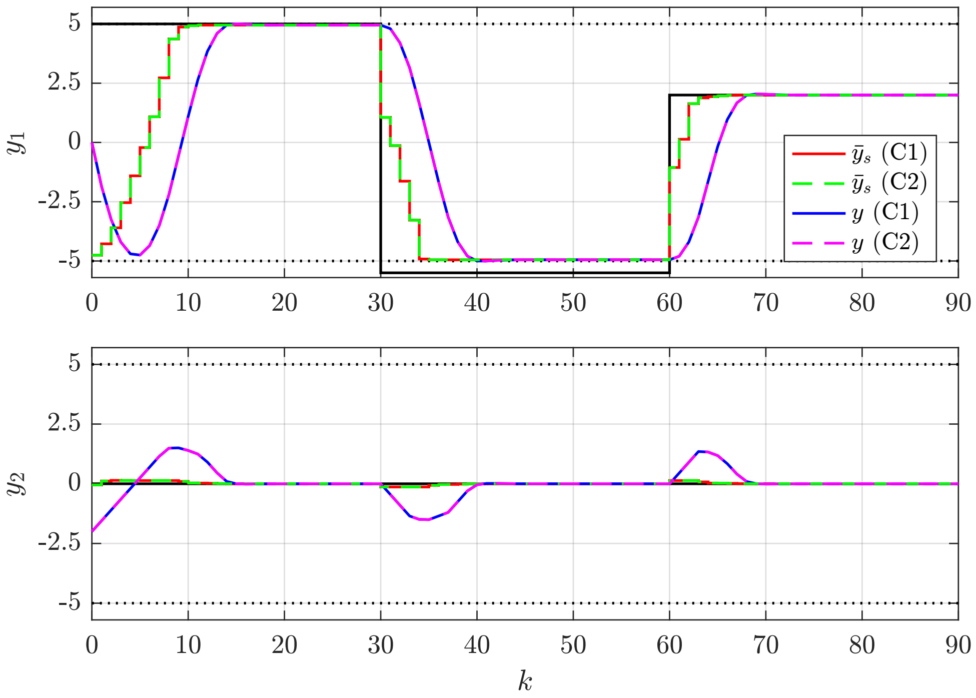

Figure 3 – Outputs of the double-integrator system controlled by C1 (proposed approach) and C2 (Limon *et al.* (2008)). Dotted black lines are the output upper/lower limits and the solid black line is the output setpoint. The remaining lines are described in the legend

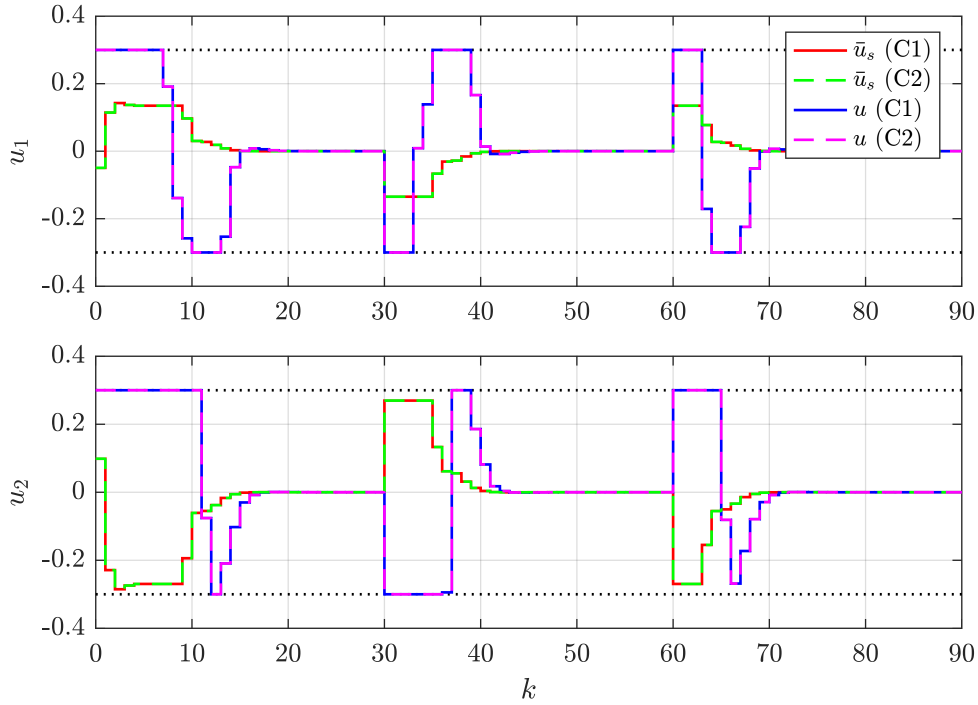

Figure 4 – Inputs of the double-integrator system controlled by C1 (proposed approach) and C2 (Limon *et al.* (2008)). Dotted black lines are the input upper/lower limits. The remaining lines are described in the legend

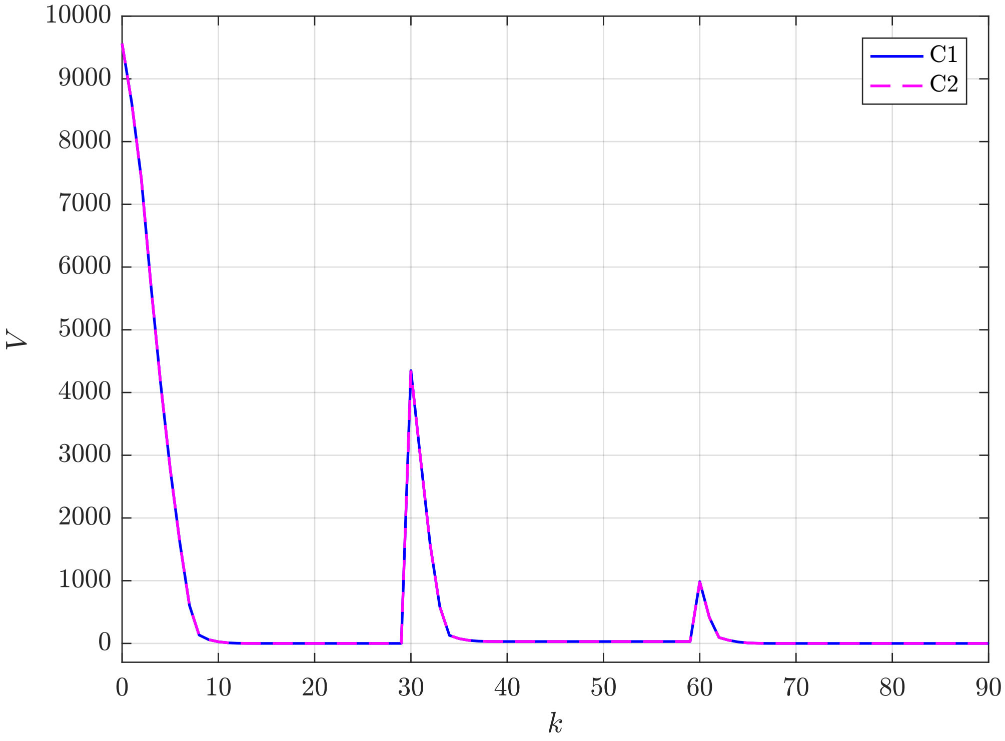

Figure 5 – Performance indexes of the double-integrator system controlled by C1 (proposed approach) and C2 (Limon *et al.* (2008))

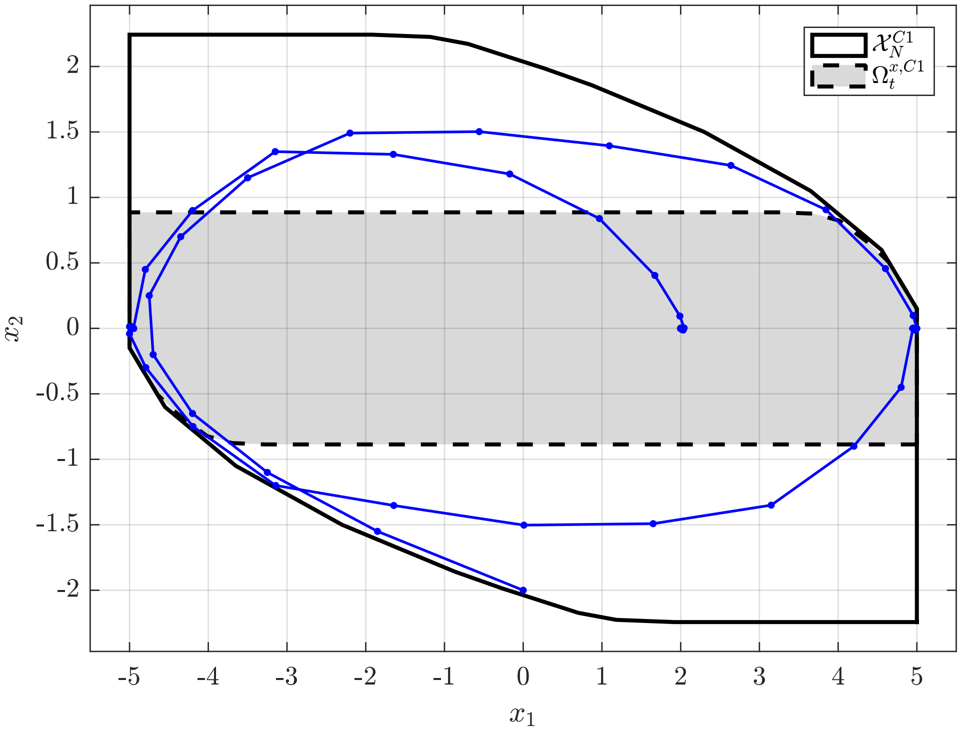

Figure 6 – Phase portrait of the double-integrator system controlled by C1 (proposed approach) and sets of feasible initial states and terminal states

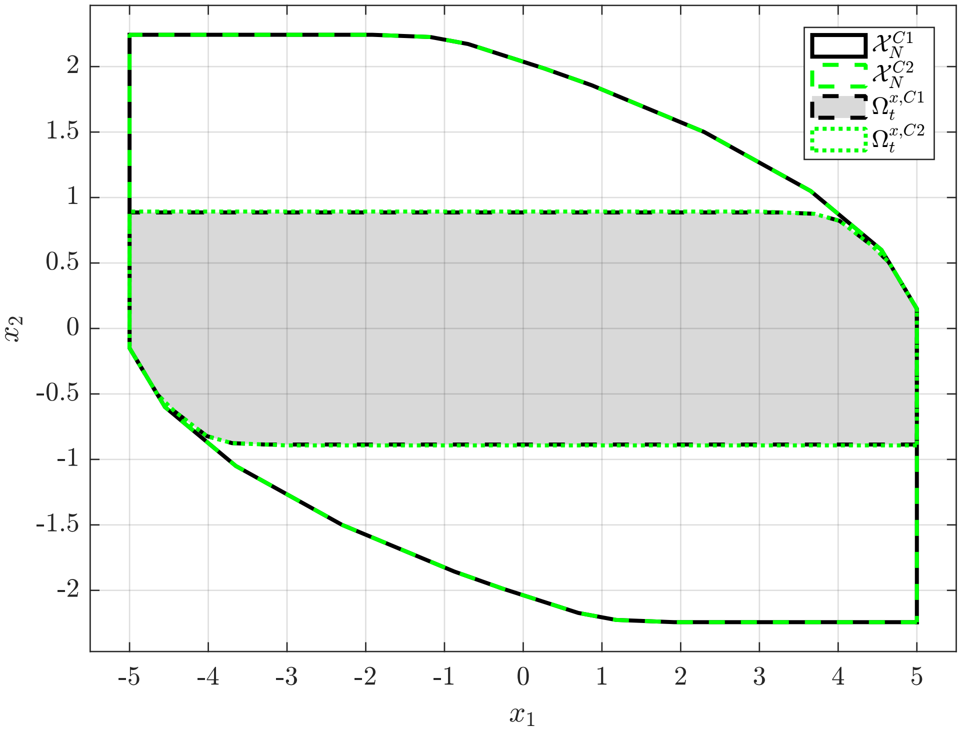

Figure 7 – Sets of feasible initial states and terminal states of controllers C1 (proposed approach) and C2 (Limon *et al.* (2008))

For C1, the set of terminal states is $\Omega_t^x = \text{Proj}_x(\Omega_t^w)$ and the set $\mathcal{W}_N$ is sliced at $u^- = (0.25, 0.1)$ to obtain the set of initial feasible states $\mathcal{X}_N$. These sets Ω_t^x and $\mathcal{X}_N$ are then written in terms of the state of the original system using (2.43) and denoted as $\Omega_t^{x,C1}$ and $\mathcal{X}_N^{C1}$, respectively. This procedure enables comparison with the sets $\mathcal{X}_N^{C2}$ and $\Omega_t^{x,C2}$ of controller C2.

Figure 6 shows the phase portrait obtained with C1 along with the sets $\Omega_t^{x,C1}$ and $\mathcal{X}_N^{C1}$. Figure 7 shows that C1 and C2 produce the same sets of terminal states and initial feasible states. Since we have used $N = 4$ for C1 and $N = 3$ for C2, one may argue that, for the same N, C2 has a larger domain of attraction $\mathcal{X}_N$ in comparison with C1. However, in terms of degrees of freedom of the optimal control problem, C2 computes not only N inputs but also an additional parameter $\theta \in \mathbb{R}^{n_u}$ that characterizes the artificial references (LIMON *et al.*, 2008), resulting in $(N+1) \times n_u$ decision variables. Conversely, C1 has only $N \times n_u$ decision variables. Therefore, since N has been chosen such that C1 and C2 have the same number of degrees of freedom, our comparison is fair.

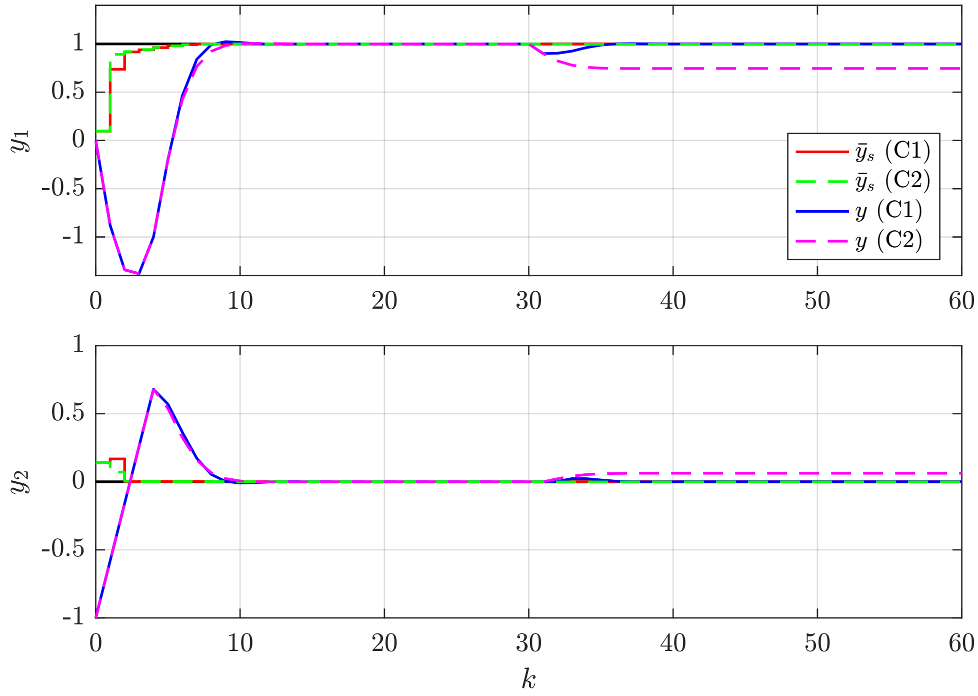

Figure 8 – Outputs of the double-integrator system controlled by C1 (proposed approach) and C2 (Limon *et al.* (2008)) in the presence of plant-model mismatch and unmeasured disturbance. The solid black line is the output setpoint and the remaining lines are described in the legend

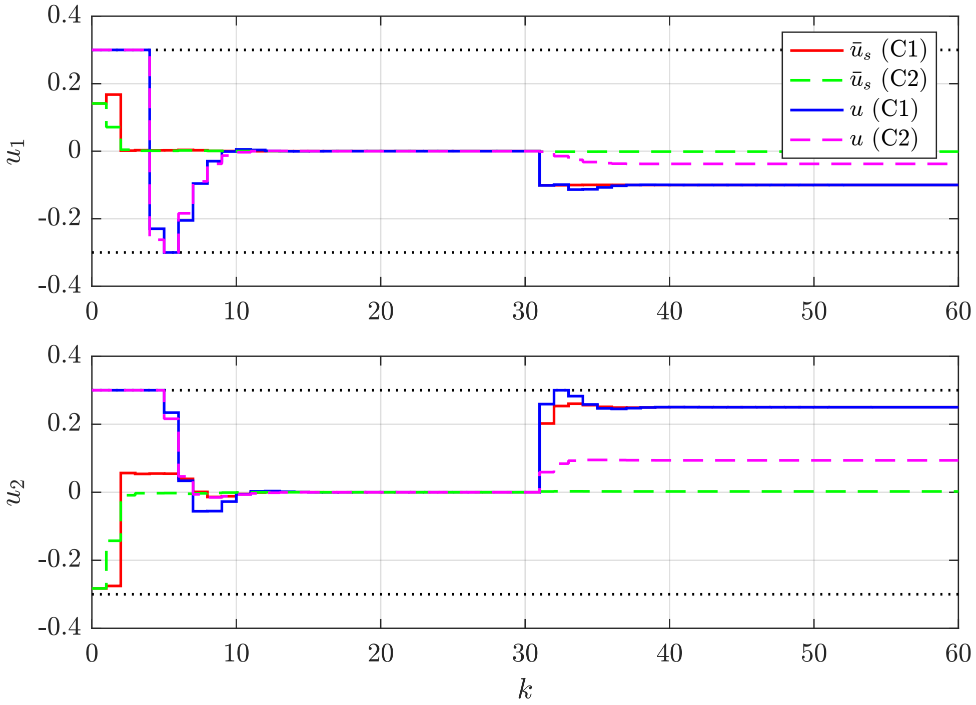

Figure 9 – Inputs of the double-integrator system controlled by C1 (proposed approach) and C2 (Limon *et al.* (2008)) in the presence of plant-model mismatch and unmeasured disturbance. Dotted black lines are the input upper/lower limits. The remaining lines are described in the legend

Controllers C1 and C2 are now compared in the presence of plant-model mismatch and unmeasured disturbances. To do so, the plant is simulated considering the following modified model:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0.4 \\ 1 & 0.4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} d$$
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Note that, in the input matrix, the elements related to u_2 were reduced in 20%, which may represent in practice an actuator fault called loss of effectiveness (COSTA *et al.*, 2021). A persistent disturbance $d = -0.1$ enters the plant at time step $k = 30$. In this simulation, the system starts at $x = (0, -1)$ and the output setpoint is fixed at $y_{sp} = (1, 0)$.

Figures 8 and 9 show system outputs and inputs, respectively. As observed, C1 steered the system to the setpoint in both cases while C2 fails to reject the disturbance, resulting in a steady-state offset. This happens because in C2 the parametrization of steady states and inputs entirely depends on the system model (LIMON *et al.*, 2008), generating artificial references that make the performance index inconsistent with zero tracking error in the presence of plant-model mismatch and unmeasured disturbances. In fact, C2 only steered the plant to the setpoint in the first part of the simulation because

$\bar{u}_s = (0, 0)$ is always a steady input for this integrating plant, which is not the case in the presence of persistent disturbances or if the plant has stable or unstable poles. Note that C1 produces consistent artificial references at steady state because the characterization of steady outputs and inputs only depends on model parameters during transients. Therefore, while C2 requires the estimation of disturbances to obtain offset-free control in this scenario, this is not necessary for C1 due to its embedded integral action characterized by the well-posed performance index that is consistent with zero tracking errors and unbiased predictions at steady state achieved by the use of an incremental model (ROSSITER, 2004).

Now, let the input increments be constrained to $\|\Delta u\|_\infty \leq 0.1$ and consider the controller C3 as a copy of C1 that takes into account this additional constraint. The first simulation scenario is reproduced here, but now the system starts at $x = (-1, -1.5)$, which is a feasible initial state for both C1 and C3. System outputs and inputs are shown in Figures 10 and 11, respectively. As expected, system convergence to the setpoint is slower with C3 in comparison to C1. On the other hand, while abrupt input movements were suppressed with C3, C1 clearly produced larger input increments, which may not be acceptable in some applications. As observed in Figure 12, C3 produced a larger performance index in comparison with C1, which was expected due to the limited convergence velocity that results from the constraint on the input increments.

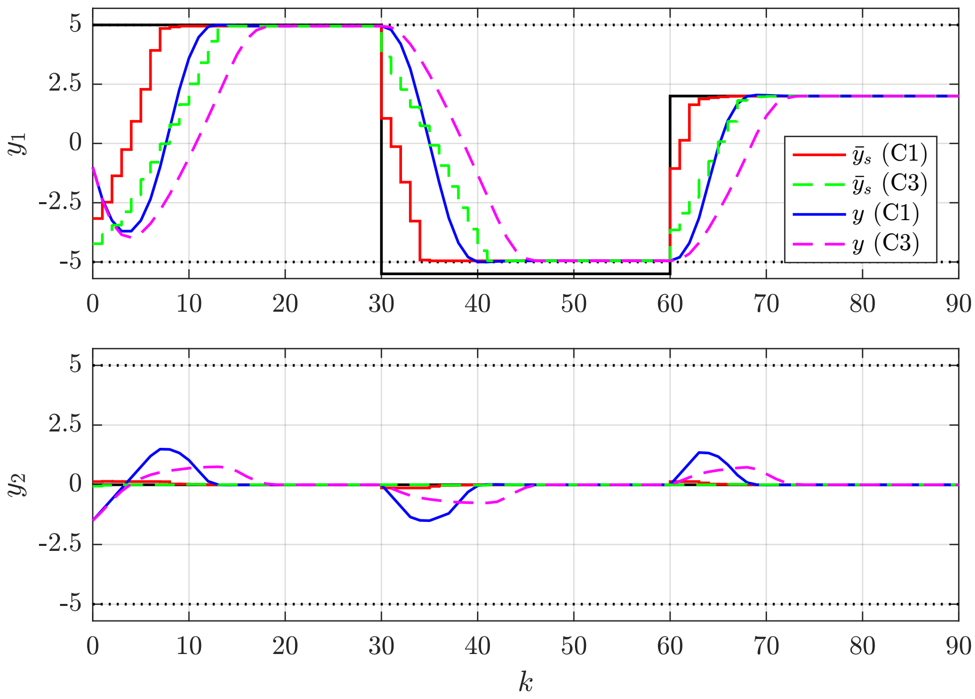

Figure 10 – Outputs of the double-integrator system controlled by C1 (unconstrained Δu) and C3 (constrained Δu). Dotted black lines are the input upper/lower limits and the solid black line is the output setpoint. The remaining lines are described in the legend

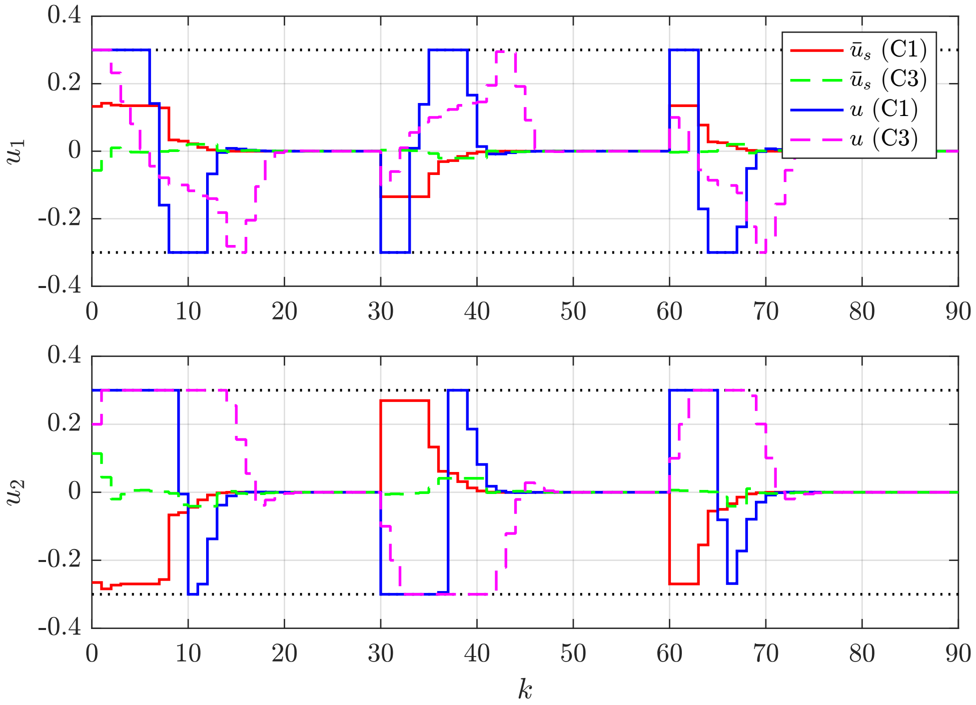

Figure 11 – Inputs of the double-integrator system controlled by C1 (unconstrained Δu) and C3 (constrained Δu). Dotted black lines are the input upper/lower limits. The remaining lines are described in the legend

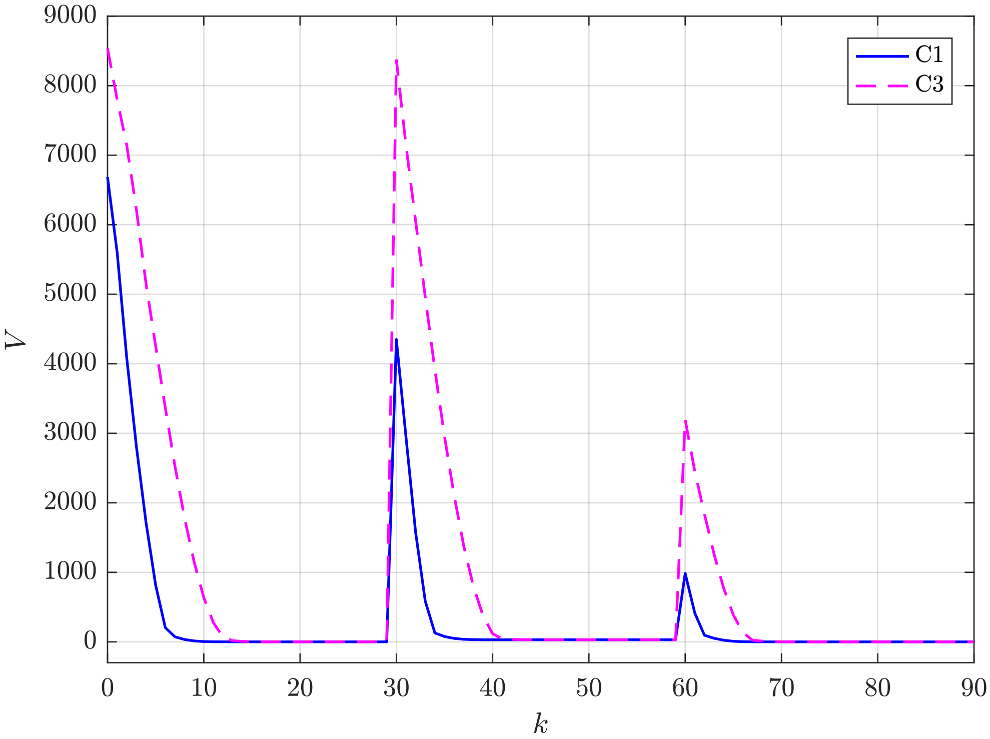

Figure 12 – Performance indexes of the double-integrator system controlled by C1 (unconstrained Δu) and C3 (constrained Δu)

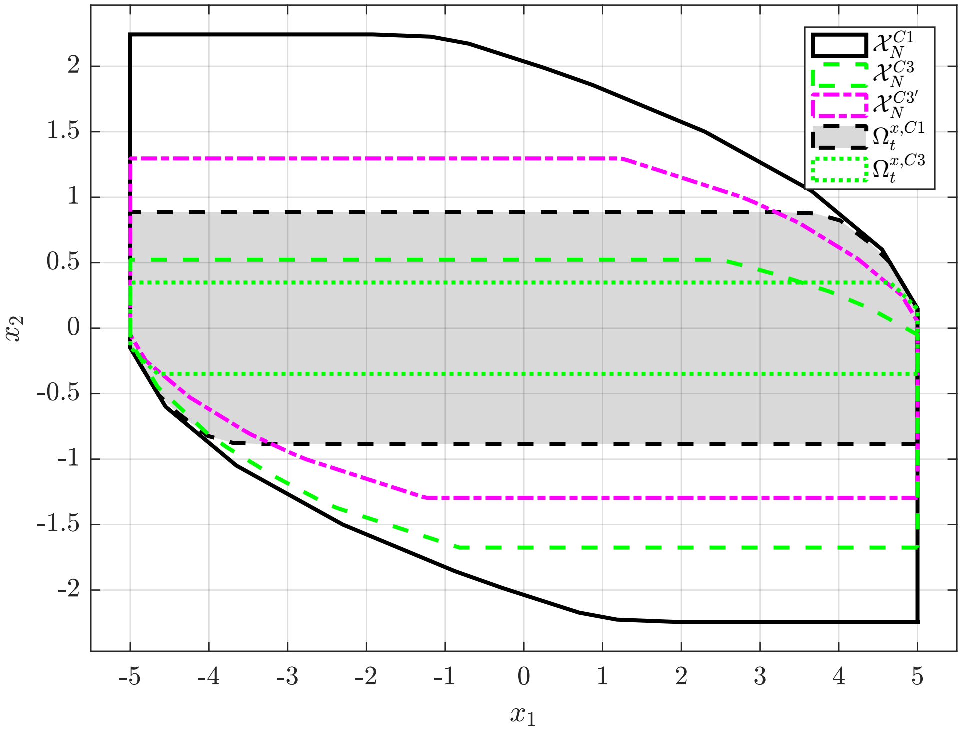

Figure 13 – Sets of feasible initial states and terminal states of controllers C1 (unconstrained Δu) and C3 (constrained Δu)

Figure 13 shows the sets of terminal states ($\Omega_t^{x,C1}$ and $\Omega_t^{x,C3}$) along with the sets of feasible initial states ($\mathcal{X}_N^{C1}$ and $\mathcal{X}_N^{C3}$), which were computed considering $u^- = (0.25, 0.1)$ for both C1 and C3. To illustrate the effect of u^-, Figure 13 also depicts $\mathcal{X}_N^{C3'}$, which denotes the set of feasible initial states computed for a predecessor input $u^- = (0, 0)$. Comparing $\Omega_t^{x,C1}$ and $\Omega_t^{x,C3}$, it easy to see that this additional constraint results in smaller invariant set. This impacts the set of initial feasible states that was substantially reduced for C3 in comparison with C1. Regarding the previously applied input u^-, while it does not affect the set of initial feasible states for the case of unconstrained input increments, different predecessor inputs may produce distinct sets of initial feasible states, which can be concluded from the comparison of $\mathcal{X}_N^{C3}$ with $\mathcal{X}_N^{C3'}$.

3.5 Output zones and input targets

3.5.1 Problem formulation

The zone control strategy is largely applied in the process industry and consists of driving or keeping system outputs inside predefined operating zones or ranges (GONZÁLEZ; ODLOAK, 2009). This is a particularly suitable tool for non-square plants with more controlled variables than manipulated ones, in which the problem of independently tracking setpoints for all controlled outputs may not be feasible, especially if these outputs are highly correlated (FERRAMOSCA *et al.*, 2010). Also, in real process applications, some input variables are required to be steered to optimizing targets in order to fulfill economic objectives and, in this case, the zone control approach can be easily applied to deal with a possible lack of degrees of freedom. Moreover, zone control is inherently robust to disturbances and plant-model mismatch (LIU; MAO; LIU, 2019) and avoids unnecessary and excessive manipulation of a control valve when controlled variables are inside their limits.

When dealing with the zone control problem (GONZÁLEZ; ODLOAK, 2009), the output setpoint y_{sp} is not fixed but rather lies in a specified range referred to as control zone, which is represented here by the set $\mathcal{Y}_{sp} \subset \mathcal{Y}$ defined as follows:

$$\mathcal{Y}_{sp} := \{y_{sp} \in \mathcal{Y} : y_{sp,min} \leq y_{sp} \leq y_{sp,max}\} \tag{3.60}$$

Note that the setpoint tracking problem is a particular case of the zone control strategy in which $y_{sp} = y_{sp,min} = y_{sp,max}$.

Let u_{sp} be the optimizing input target provided by an RTO layer. A more general way of accommodating steady-state objectives is by using the so-called offset cost function (FERRAMOSCA *et al.*, 2009), which is denoted here as $V_O(\bar{z}_s, z_{sp})$, with $V_O : \mathbb{R}^{2(n_y+n_u)} \to \mathbb{R}$ and $z_{sp} = (y_{sp}, u_{sp})$. In terms of V_O, the performance index is written as follows:

$$\begin{aligned} V_N^{zt}(w, \Delta\mathbf{u}, z_{sp}) = \sum_{j=0}^{N-1} \left(\|y(j) - \bar{y}_s\|_{Q_y}^2 + \|u(j) - \bar{u}_s\|_{Q_u}^2 + \|\Delta u(j)\|_R^2 \right) \\ + \|x_d(N)\|_P^2 + V_O(\bar{z}_s, z_{sp}) \end{aligned} \tag{3.61}$$

Assumption 3. The offset cost function $V_O(\bar{z}_s, z_{sp})$ is subdifferentiable and convex w.r.t $(\bar{z}_s, z_{sp})$. If $\bar{z}_s = z_{sp}$, then $V_O(\bar{z}_s, z_{sp}) = 0$. Otherwise, $V_O(\bar{z}_s, z_{sp}) > 0$.

In an industrial application of the MPC with OPOM (MARTIN; ZANIN; ODLOAK, 2019) and in many other works (GONZÁLEZ; ODLOAK, 2009; MARTINS *et al.*, 2013; MARTINS; ODLOAK, 2016), the offset cost function (equivalently viewed as the penalization of slack variables in these studies) is simply defined in terms of squared weighted Euclidean norms, i.e. $V_O(\bar{z}_s, z_{sp}) = \|\bar{y}_s - y_{sp}\|_{S_y}^2 + \|\bar{u}_s - u_{sp}\|_{S_u}^2$. The use of different norms (e.g. 1-norm and ∞-norm) is addressed in Ferramosca *et al.* (2010). Conditions under which the choice of the offset cost function provides the best possible performance index are studied in Ferramosca *et al.* (2009).

Now, the proposed dual-mode MPC with OPOM extended for the case of output zone control and input target tracking is based on the control problem $P_N^{zt}(w, \mathcal{Y}_{sp}, u_{sp})$, which is defined below for a given extended state $w = (x, u^-)$, control zone $\mathcal{Y}_{sp}$ and input target u_{sp}:

$$(\Delta\mathbf{u}^*, y_{sp}^*) = \arg\min_{\Delta\mathbf{u}, y_{sp}} \quad V_N^{zt}(w, \Delta\mathbf{u}, z_{sp}) \tag{3.62}$$

subject to

$$(x(0), u(-1)) = w \tag{3.63}$$

$$x(j+1) = Ax(j) + B\Delta u(j), \quad j \in \mathbb{I}_{0:N-1} \tag{3.64}$$

$$y(j) = Cx(j), \quad j \in \mathbb{I}_{0:N-1} \tag{3.65}$$

$$u(j) = u(j-1) + \Delta u(j), \quad j \in \mathbb{I}_{0:N-1} \tag{3.66}$$

$$y(j) \in \mathcal{Y}, \quad j \in \mathbb{I}_{0:N-1} \tag{3.67}$$

$$u(j) \in \mathcal{U}, \quad j \in \mathbb{I}_{0:N-1} \tag{3.68}$$

$$\Delta u(j) \in \mathcal{U}_\Delta, \quad j \in \mathbb{I}_{0:N-1} \tag{3.69}$$

$$y_{sp} \in \mathcal{Y}_{sp} \tag{3.70}$$

$$(\bar{y}_s, \bar{u}_s) = M(x(N), u(N-1)) \tag{3.71}$$

$$(x(N), u(N-1)) \in \Omega_t^w \tag{3.72}$$

In the same manner as problem $P_N(w, y_{sp})$, the solution of $P_N^{zt}(w_k, \mathcal{Y}_{sp,k}, u_{sp,k})$ at a given time step k results in the sequence of control increments $\Delta\mathbf{u}_k^*$, whose first element $\Delta u^*(0|k)$ is applied to the system. Hence, the control law $\Delta u(k) := \kappa_N^{zt}(w_k, \mathcal{Y}_{sp,k}, u_{sp,k})$ is implicitly produced by the receding horizon implementation.

3.5.2 Recursive feasibility and convergence analyses

Recursive feasibility and convergence properties of this extended controller are assessed in the following Lemma and Theorem, respectively.

Lemma 4 (Recursive feasibility)**.** *Let Assumption 2 hold and consider a time step k with an extended state $w_k \in \mathcal{W}_N$ and control zone $\mathcal{Y}_{sp,k}$. Then, the feasibility of problem $P_N^{zt}(w_k, \mathcal{Y}_{sp,k}, u_{sp,k})$ at time step k implies that $P_N^{zt}(w_{k+j}, \mathcal{Y}_{sp,k+j}, u_{sp,k+j})$ will remain feasible at any subsequent time step $k+j$, $\forall j \in \mathbb{I}_{\geq 1}$.*

Proof. Let $(\Delta\mathbf{u}_k^*, y_{sp,k}^*)$ denote the solution of problem $P_N^{zt}(w_k, \mathcal{Y}_{sp,k}, u_{sp,k})$ at time step k. After the application of $\Delta u^*(0|k)$ into the system, we move to time step $k+1$, at which $x(0|k+1) = x(k+1) = x(1|k)$. Consider a suboptimal candidate solution to problem $P_N^{zt}(w_{k+1}, \mathcal{Y}_{sp,k+1}, u_{sp,k+1})$ denoted as $(\Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$, in which $\Delta\tilde{\mathbf{u}}_{k+1}$ is given by (3.43). Now, observe that choosing $\tilde{y}_{sp,k+1} = y_{sp,k}^*$ clearly satisfies (3.70) when $y_{sp,k}^* \in \mathcal{Y}_{sp,k} \cap \mathcal{Y}_{sp,k+1}$. However, since the terminal stability constraint (3.72) does not depend on y_{sp}, there is no conflict with constraint (3.70) and, consequently, any $\tilde{y}_{sp,k+1} \in \mathcal{Y}_{sp,k+1}$ can be chosen so as to satisfy (3.70). Since the input target u_{sp} only appears in the objective function, it does not affect the feasibility of problem $P_N^{zt}(w, \mathcal{Y}_{sp}, u_{sp})$. The remaining constraints are satisfied by the same arguments presented in Lemma 1.

Therefore, by induction, problem $P_N^{zt}(w_{k+j}, \mathcal{Y}_{sp,k+j}, u_{sp,k+j})$ will remain feasible at every time step $k+j$, $\forall j \in \mathbb{I}_{\geq 1}$. □

Corollary 1. *Given the same sets $\mathcal{Y}$, $\mathcal{U}$ and $\mathcal{U}_\Delta$ of system constraints and the same terminal set Ω_t^w, then problems $P_N(w, y_{sp})$ and $P_N^{zt}(w, \mathcal{Y}_{sp}, u_{sp})$ have the same set $\mathcal{W}_N$ of feasible initial augmented states.*

Proof. This is a direct result of the fact that the terminal constraint does not depend on y_{sp} in both problems $P_N(w, y_{sp})$ and $P_N^{zt}(w, \mathcal{Y}_{sp}, u_{sp})$ and that the inclusion of control zones and input targets does not affect the feasibility of problem $P_N^{zt}(w, \mathcal{Y}_{sp}, u_{sp})$, as shown in Lemma 4. □

Theorem 2 (Closed-loop convergence)**.** *Let Assumption 2 hold and consider a control zone $\mathcal{Y}_{sp}$ and an input target u_{sp}. Then, for any feasible initial augmented state $w_0 \in \mathcal{W}_N$, the system* (3.3)-(3.4) *with inputs computed through control law $\kappa_N^{zt}(w_k, \mathcal{Y}_{sp}, u_{sp})$ is admissibly steered to an operating point $\bar{z}_s = (\bar{y}_s, \bar{u}_s) \in \bar{\mathcal{Z}}_s^*$, in which $\bar{\mathcal{Z}}_s^*$ is the set of all possible optimal operating points defined as follows:*

$$\bar{\mathcal{Z}}_s^* = \left\{ \bar{z}_s^* : (\bar{z}_s^*, z_{sp}^*) = \arg \min_{\substack{\bar{z}_s \in \bar{\mathcal{Z}}_s \\ y_{sp} \in \mathcal{Y}_{sp}}} V_O(\bar{z}_s, z_{sp}) \right\}$$

Moreover, if $(\mathcal{Y}_{sp} \times u_{sp}) \cap \bar{\mathcal{Z}}_s \neq \emptyset$, then $\bar{y}_s = y_{sp} \in \mathcal{Y}_{sp}$ and $\bar{u}_s = u_{sp}$.

Proof. Consider a given initial augmented state $w_k \in \mathcal{W}_N$ at time step k at which problem $P_N^{zt}(w_k, \mathcal{Y}_{sp}, u_{sp})$ is feasible with optimal solution denoted as $(\Delta\mathbf{u}_k^*, y_{sp,k}^*)$ and optimal value function $V_N^{zt*}(w_k, \Delta\mathbf{u}_k^*, y_{sp,k}^*)$. Then, by the receding horizon principle, only the first input movement is injected into the system and we move to time step $k+1$, at which a solution to problem $P_N^{zt}(w_{k+1}, \mathcal{Y}_{sp}, u_{sp})$ is guaranteed to exist by virtue of Lemma 4.

At time step $k+1$, consider a suboptimal solution to problem $P_N^{zt}(w_{k+1}, \mathcal{Y}_{sp}, u_{sp})$ denoted as $(\Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$, in which $\Delta\tilde{\mathbf{u}}_{k+1}$ is given in (3.43) and $\tilde{y}_{sp,k+1} = y_{sp,k}^*$. As already shown in Lemma 4, $(\Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$ is feasible. Let $\tilde{V}_N^{zt}(w_{k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, \tilde{z}_{sp,k+1})$ denote the corresponding performance criterion with $\tilde{z}_{sp,k+1} = (\tilde{y}_{sp,k+1}, u_{sp})$.

Since this suboptimal solution produces the same artificial references $(\bar{y}_{s,k}, \bar{u}_{s,k})$, the steps presented in the proof of Theorem 1 can be followed to show that:

$$V_N^{zt*}(w_{k+1}, \Delta\mathbf{u}_{k+1}^*, z_{sp,k+1}^*) \leq V_N^{zt*}(w_k, \Delta\mathbf{u}_k^*, z_{sp,k}^*), \quad \forall k \in \mathbb{N}$$

Therefore, by the same arguments used in Theorem 1, we can conclude that $\lim_{k\to\infty}\|y(k) - \bar{y}_{s,k}\| = 0$, $\lim_{k\to\infty}\|u(k) - \bar{u}_{s,k}\| = 0$ and $\lim_{k\to\infty}\|\Delta u(k)\| = 0$. This implies that the system converges to an operating point $\bar{z}_{s,\infty} = \lim_{k\to\infty}(\bar{y}_{s,k}, \bar{u}_{s,k})$, corresponding to a performance index given as $V_N^{zt*}(w_\infty, \Delta\mathbf{u}^*, z_{sp}^*) = V_O(\bar{z}_{s,\infty}, z_{sp}^*)$.

Now, let us show that $\bar{z}_{s,\infty} \in \bar{\mathcal{Z}}_s^*$. To prove by contradiction, assume that $\bar{z}_{s,\infty} \notin \bar{\mathcal{Z}}_s^*$. Then, there exists a $\bar{z}_s^* \in \bar{\mathcal{Z}}_s^*$ such that $V_O(\bar{z}_s^*, z_{sp}^*) < V_O(\bar{z}_{s,\infty}, z_{sp}^*)$. By the definition of $\bar{\mathcal{Z}}_s$, there is also a w^* such that $\bar{z}_s^* = Mw^*$ and $w^* \in \Omega_t^w$. Consider a $w^+ = (x^+, u)$ given as $w^+ = \gamma w_\infty + (1-\gamma)w^*$ in which $\gamma \in (0,1)$. In terms of y_s, x_d and u, this convex combination is:

$$\begin{bmatrix} y_s^+ \\ x_d^+ \\ u \end{bmatrix} = \gamma \begin{bmatrix} \bar{y}_{s,\infty} \\ 0 \\ \bar{u}_{s,\infty} \end{bmatrix} + (1-\gamma) \begin{bmatrix} y_s^* \\ x_d^* \\ u^* \end{bmatrix}$$

Since Ω_t^w is a convex polyhedron and both $w_\infty \in \Omega_t^w$ and $w^* \in \Omega_t^w$, it follows that $w^+ \in \Omega_t^w$ and, consequently, $(y_s^+ + C_d x_d^+) \in \mathcal{Y}$ for any $\gamma \in (0,1)$. Similarly, any $\gamma \in (0,1)$ produces $u \in \mathcal{U}$ because $\bar{u}_{s,\infty} \in \lambda\mathcal{U}$ and $u^* \in \mathcal{U}$. Observing that the input increment needed to steer w_∞ to w^+ corresponds to $\Delta u = u - \bar{u}_{s,\infty} = (1-\gamma)(u^* - \bar{u}_{s,\infty})$, there exists a sufficiently large γ_{min} such that $\Delta u \in \mathcal{U}_\Delta$ for any $\gamma \in [\gamma_{min}, 1)$.

Now, taking $\gamma \in [\gamma_{min}, 1)$, we can build a sequence of input movements given as $\Delta\tilde{\mathbf{u}} = \{\Delta u, Kx_d^+, \ldots, KF^{N-2}x_d^+\}$, which is feasible due to the positive invariance of Ω_t^w. By following a similar procedure to the one used to obtain (3.46) in Lemma 2, it can be shown that the performance index associated with $(\Delta\tilde{\mathbf{u}}, z_{sp}^*)$ can be written as follows:

$$V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*) = \|\bar{y}_{s,\infty} - \bar{y}_s^+\|_{Q_y}^2 + \|u - \bar{u}_s^+\|_{Q_u}^2 + \|\Delta u\|_R^2 + \|x_d^+\|_P^2 + V_O(\bar{z}_s^+, z_{sp}^*) \quad (3.73)$$

in which $\bar{z}_s^+ = (\bar{y}_s^+, \bar{u}_s^+) = M(x^+, u)$, with $x^+ = (y_s^+, x_d^+)$, $y_s^+ = \bar{y}_{s,\infty} + B_s\Delta u$, $x_d^+ = B_d\Delta u$ and $u = \bar{u}_{s,\infty} + \Delta u$.

By following the same steps used to obtain (3.47)-(3.50), it can be shown that (3.73) can be rewritten as follows:

$$V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*) = \|\Delta u\|_{(\tilde{Q}_y + \tilde{Q}_u + R + \tilde{P})}^2 + V_O(\bar{z}_s^+, z_{sp}^*) \quad (3.74)$$

The substitution of $\Delta u = (1-\gamma)(u^* - \bar{u}_{s,\infty})$ into (3.74) results:

$$V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*) = (1-\gamma)^2 \|w_\infty - w^*\|_{Q_w}^2 + V_O(\bar{z}_s^+, z_{sp}^*) \quad (3.75)$$

with $Q_w = \bar{I}_u^T \left(\tilde{Q}_y + \tilde{Q}_u + R + \tilde{P}\right) \bar{I}_u$ and $\bar{I}_u = \begin{bmatrix} 0_{n_u \times n_y} & 0_{n_u \times n_d} & I_{n_u} \end{bmatrix}$.

Note that $V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*) = V_O(\bar{z}_{s,\infty}, z_{sp}^*) = V_N^{zt}(w_\infty, \Delta\mathbf{u}^*, z_{sp}^*)$ for $\gamma = 1$.

Now, following the ideas presented in Ferramosca *et al.* (2009), let $\partial V_O(\bar{z}_s^+, z_{sp}^*)$ denote the subdifferential of V_O at $(\bar{z}_s^+, z_{sp}^*)$, then the partial of $V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*)$ about γ is:

$$\frac{\partial V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*)}{\partial\gamma} = -2(1-\gamma)\|w_\infty - w^*\|_{Q_w}^2 + g^T(\bar{z}_{s,\infty} - \bar{z}_s^*)$$

in which $g^T \in \partial V_O(\bar{z}_s^+, z_{sp}^*)$. For $\gamma = 1$, we have:

$$\left.\frac{\partial V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*)}{\partial\gamma}\right|_{\gamma=1} = \bar{g}^T(\bar{z}_{s,\infty} - \bar{z}_s^*)$$

in which $\bar{g}^T \in \partial V_O(\bar{z}_{s,\infty}, z_{sp}^*)$ and $\partial V_O(\bar{z}_{s,\infty}, z_{sp}^*)$ denotes the subdifferential of V_O at $(\bar{z}_{s,\infty}, z_{sp}^*)$. From the interpretation of the subdifferential in the context of minorization by affine functions (HIRIART-URRUTY; LEMARÉCHAL, 2001, p. 167), the following inequality holds:

$$V_O(\bar{z}_s^*, z_{sp}^*) \geq V_O(\bar{z}_{s,\infty}, z_{sp}^*) + \bar{g}^T \left(\bar{z}_s^* - \bar{z}_{s,\infty}\right)$$

Given that $V_O(\bar{z}_s^*, z_{sp}^*) < V_O(\bar{z}_{s,\infty}, z_{sp}^*)$ as $\bar{z}_s^* \in \bar{\mathcal{Z}}_s^*$ and $\bar{z}_{s,\infty} \notin \bar{\mathcal{Z}}_s^*$, it follows that:

$$\begin{aligned} \left.\frac{\partial V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*)}{\partial\gamma}\right|_{\gamma=1} &= \bar{g}^T \left(\bar{z}_{s,\infty} - \bar{z}_s^*\right) \\ &\geq V_O(\bar{z}_{s,\infty}, z_{sp}^*) - V_O(\bar{z}_s^*, z_{sp}^*) > 0 \end{aligned}$$

Thus, since $V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*)$ is positively related to γ, there exists a $\gamma \in [\gamma_{min}, 1)$ that produces a smaller value of $V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*)$ in comparison to the value it takes with $\gamma = 1$. However, since $V_N^{zt}(w_\infty, \Delta\tilde{\mathbf{u}}, z_{sp}^*) = V_N^{zt}(w_\infty, \Delta\mathbf{u}^*, z_{sp}^*)$ for $\gamma = 1$, this contradicts the optimality of $V_N^{zt}(w_\infty, \Delta\mathbf{u}^*, z_{sp}^*)$, and thus $\bar{z}_{s,\infty} \in \bar{\mathcal{Z}}_s^*$.

Now, observe that y_{sp} is a decision variable that appears directly in the offset cost function and it is only constrained to $y_{sp} \in \mathcal{Y}_{sp}$. Then, for any given $\bar{z}_{s,k} \in \bar{\mathcal{Z}}_s$ at time step k, an optimal y_{sp}^* is also a minimizer of $V_O(\bar{z}_{s,k}, z_{sp}^*)$. Consequently, $V_O^*(\bar{z}_{s,\infty}, z_{sp}^*)$ corresponds to the smallest value that the offset cost function takes for a given control zone $\mathcal{Y}_{sp}$ and input target u_{sp}.

To complete the proof, let us analyze the case in which $(\mathcal{Y}_{sp} \times u_{sp}) \cap \bar{\mathcal{Z}}_s \neq \emptyset$. Observe that, in this circumstance, there exists a $\bar{z}_{s,\infty} \in \bar{\mathcal{Z}}_s^*$ such that $\bar{z}_{s,\infty} = z_{sp}^*$, which results in $V_O^*(\bar{z}_{s,\infty}, z_{sp}^*) = 0$ by virtue of Assumption 3. Therefore, if $(\mathcal{Y}_{sp} \times u_{sp}) \cap \bar{\mathcal{Z}}_s \neq \emptyset$, the system converges to an operating point given by $\bar{y}_s = y_{sp} \in \mathcal{Y}_{sp}$ and $\bar{u}_s = u_{sp}$. □

3.6 Case study 2: unstable reactor system

In this section, the proposed MPC with zone control and input targets is applied to a continuous stirred-tank reactor (CSTR), in which an exothermic diabatic irreversible first-order reaction ($A \rightarrow B$) takes place. Assuming constant physical parameters, perfect mixing and static holdup, the dynamics of this system are obtained from material and energy balances, as described in Russo and Bequette (1997). A model nondimensionalization is performed with representative values of dimensionless model variables and parameters given in Russo and Bequette (1997), resulting in the following set of nonlinear ordinary differential equations:

$$\frac{dx_1}{d\tau} = q\left(x_{1f} - x_1\right) - \phi x_1 \exp\left(\frac{x_2}{1 + x_2/\gamma}\right)$$

$$\frac{dx_2}{d\tau} = q\left(x_{2f} - x_2\right) - \delta\left(x_2 - x_3\right) + \beta\phi x_1 \exp\left(\frac{x_2}{1 + x_2/\gamma}\right)$$

$$\frac{dx_3}{d\tau} = \frac{q_c\left(x_{3f} - x_3\right)}{\delta_1} + \frac{\delta\left(x_2 - x_3\right)}{\delta_1\delta_2}$$

Table 1 – Model parameters of the CSTR system

Parameter	β	γ	δ	δ_1	δ_2	ϕ	x_{1f}	x_{2f}	x_{3f}
Value	8.0	20	0.3	0.1	0.5	0.072	1	0	-1

Values of model parameters are given in Table 1, while model variables are described in Table 2 together with steady-state values at which this model is linearized. These values correspond to an unstable equilibrium point of the system (MARTINS, 2014).

Table 2 – Variables of the CSTR system

Symbol	Dimensionless variable	Linearization point
x_1	concentration of reactant A	0.5528
x_2	reactor temperature	0.2517
x_3	jacket temperature	0
q	reactor feed flow rate	1
q_c	jacket flow rate	1.6510

Aiming at reproducing a common situation encountered in chemical processes, Martins and Odloak (2016) included time delays in this system, resulting in the following model:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \\ y_3(s) \end{bmatrix} = \begin{bmatrix} \frac{0.4472(s+22.59)(s+0.5798)e^{-0.05s}}{(s+22.58)(s+0.8925)(s-0.6222)} & \frac{1.0365}{(s+22.58)(s+0.8925)(s-0.6222)} \\ \frac{-2.7517(s+22.51)(s+0.7572)e^{-0.05s}}{(s+22.58)(s+0.8925)(s-0.6222)} & \frac{-3(s+1.809)}{(s+22.58)(s+0.8925)(s-0.6222)} \\ \frac{-16.51(s+0.7572)}{(s+22.58)(s+0.8925)(s-0.6222)} & \frac{-10(s+0.8374)(s-0.4926)e^{-0.2s}}{(s+22.58)(s+0.8925)(s-0.6222)} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \tag{3.76}$$

in which the outputs represent dimensionless states x_1, x_2 and x_3 and the inputs u_1 and u_2 correspond to q and q_c, respectively. A state-space realization of model (3.76) has been discretized using the zero-order hold method with sampling time $T_s = 0.05$. The resulting discrete-time state-space model was used to simulate the plant and to build the new OPOM through the formulas presented in Chapter 2. Hereafter, system outputs and inputs represent deviation variables with respect to the linearization point given in Table 2.

The system is subject to constraints on inputs and input increments whose limits are given as $u_{min} = \begin{bmatrix}-1 & -1.5\end{bmatrix}^T$, $u_{max} = \begin{bmatrix}1 & 1.5\end{bmatrix}^T$ and $\Delta u_{max} = -\Delta u_{min} = \begin{bmatrix}0.5 & 0.5\end{bmatrix}^T$. An LQR has been used to design the terminal control law with $Q_{LQR} = I_3$ and $R_{LQR} = I_2$ and the invariant set for tracking has been computed using $\lambda = 0.99$. The offset cost function is simply selected as $V_O(\bar{z}_s, z_{sp}) = \|\bar{y}_s - y_{sp}\|^2_{S_y} + \|\bar{u}_s - u_{sp}\|^2_{S_u}$ and the tuning parameters of the proposed MPC are $N = 5$, $Q_y = I_3$, $Q_u = 0 \times I_2$, $R = I_2$, $S_y = 10^3 \times I_2$ and $S_u = 10^2 \times \text{diag}\left(\begin{bmatrix}0 & 5\end{bmatrix}\right)$.

The simulation starts with $y = \begin{bmatrix}-0.1360 & 1.1074 & 0.3224\end{bmatrix}^T$ and $u^- = \begin{bmatrix}0.3 & -0.0595\end{bmatrix}^T$ and, initially, the objective is to steer the system to the origin, i.e. $y_{sp} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$. At time step $k = 100$, the zone control mode is activated with $y_{sp,min} = \begin{bmatrix}-0.20 & 1.5 & -1\end{bmatrix}^T$ and $y_{sp,max} = \begin{bmatrix}-0.15 & 1.5 & 1\end{bmatrix}^T$. Note that, while the concentration of reactant A and the jacket temperature are controlled within zones, the reactor temperature still has a setpoint. Initially, $u_{sp} = \begin{bmatrix}0 & 0\end{bmatrix}^T$, which is changed to $u_{sp} = \begin{bmatrix}0 & -0.5\end{bmatrix}^T$ at $k = 200$. However, observe that only u_2 actually has a target due to the way we have selected S_u.

As depicted in Figure 14, system outputs are driven either to their setpoints or into the control zones. Note that, as long as $\bar{y}_s \in \mathcal{Y}_{sp}$, the output setpoint is chosen such that $y_{sp} = \bar{y}_s$, which is equivalent to have a reduced offset cost function given as $V_O(\bar{z}_s, z_{sp}) = \|\bar{u}_s - u_{sp}\|^2_{S_u}$. This remaining term of the offset cost function vanishes when the jacket flow rate is steered to its target, as shown in Figure 15. Input limits were achieved for a few moments during transients and piece-wise constant lines allow one to observe that constraints on input increments were also satisfied. Figure 16 shows the monotonically decreasing behavior of the performance index after changes on setpoint, zone control and input target.

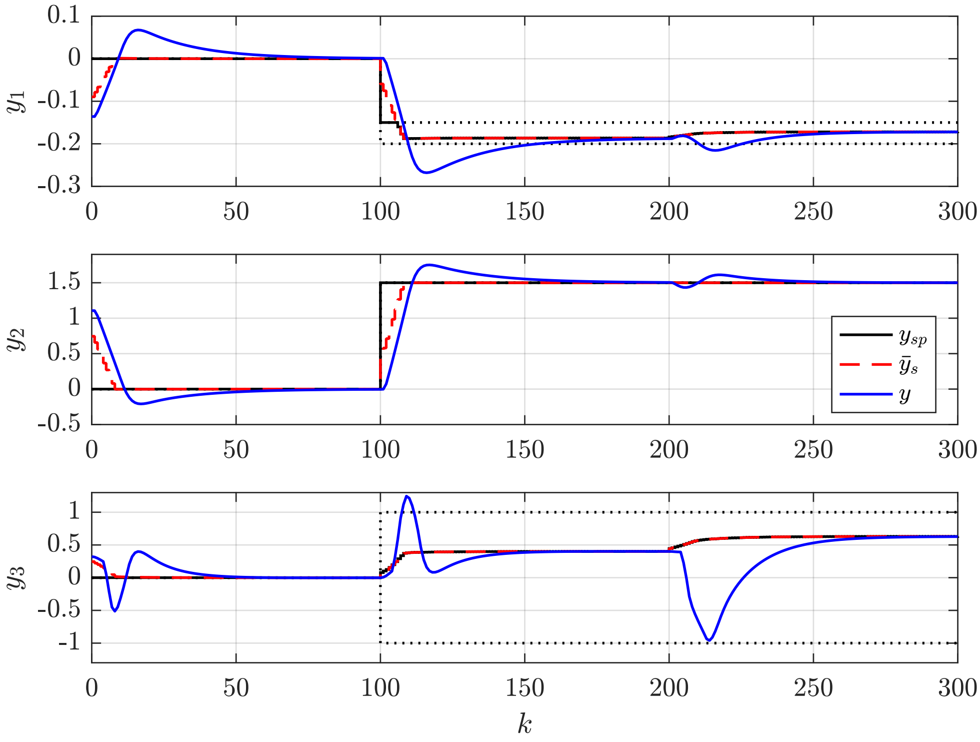

Figure 14 – Outputs of the unstable reactor system. Dotted black lines are the output control zones. The remaining lines are described in the legend

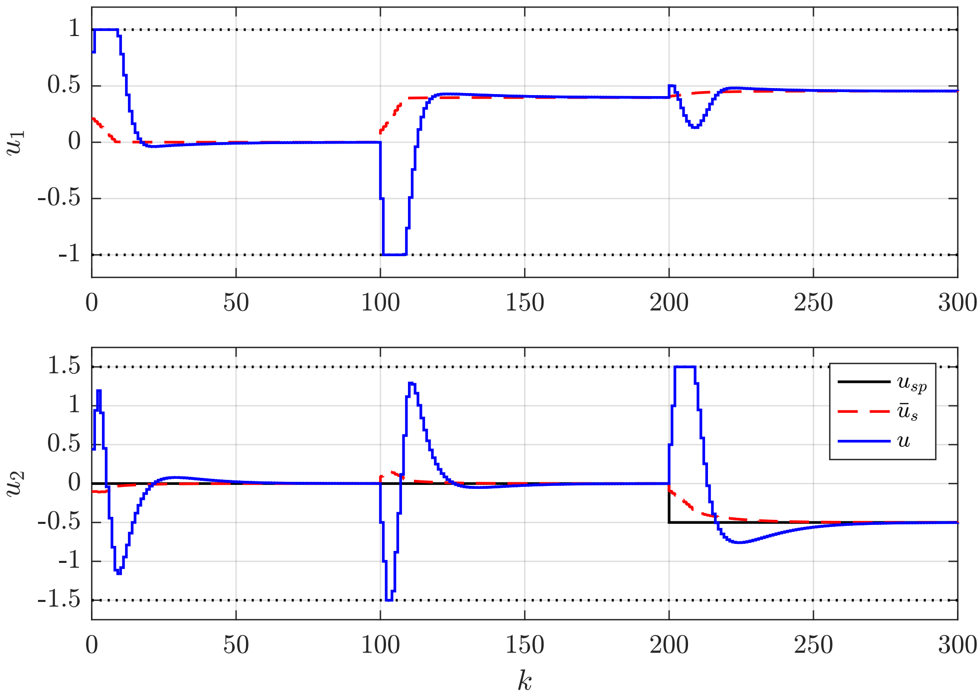

Figure 15 – Inputs of the unstable reactor system. Dotted black lines are the input upper/lower limits. The remaining lines are described in the legend

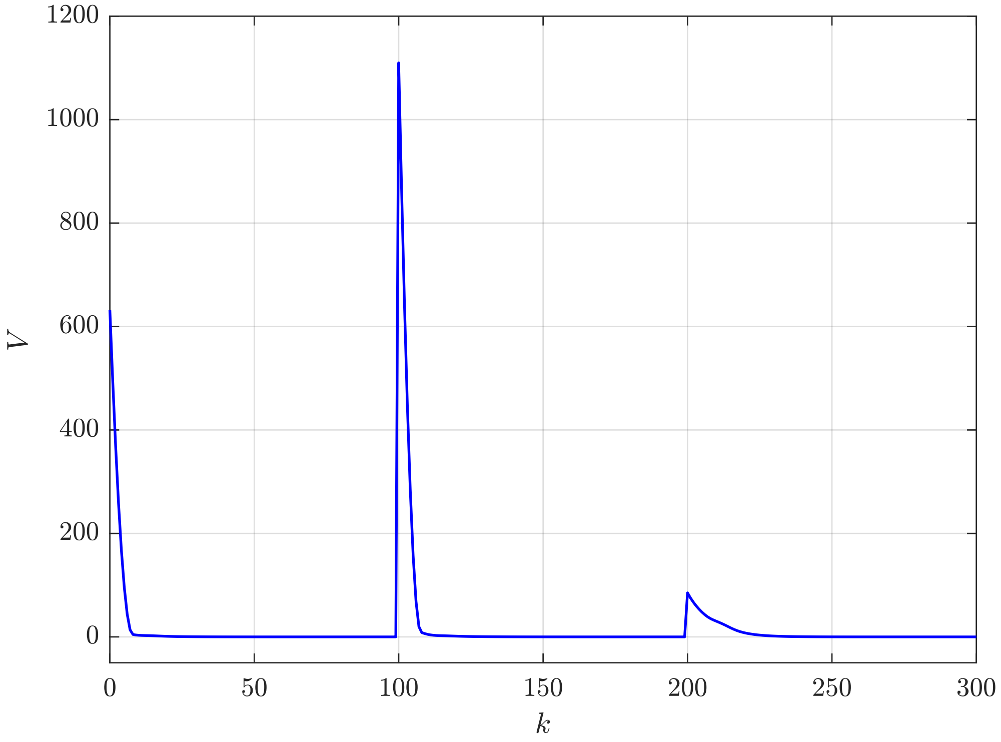

Figure 16 – Performance index of the unstable reactor system

3.7 Conclusion

This chapter proposes combining the approaches presented in Rodrigues and Odloak (2003b) and Limon *et al.* (2008) to formulate a dual-mode MPC with OPOM. Here, we adopt a performance index based on artificial references, which is shown to be equivalent to the use of slack variables, employed in Rodrigues and Odloak (2003b). A simple characterization of artificial references that is based only on terminal states and inputs is presented. Unlike in Limon *et al.* (2008), the proposed method avoids the parametrization of system steady states and inputs. Moreover, since artificial references do not depend on model parameters at steady state, they converge to a consistent real plant equilibrium even in the presence of plant-model mismatch or constant unmeasured disturbances. Explicitly considering constraints on input increments, a new characterization of an invariant set for tracking is presented. The combination of all these ingredients results in a dual-mode MPC strategy with embedded integral action and guaranteed recursive feasibility and stability.

A comparison between the MPC with OPOM and the one proposed by Limon *et al.* (2008) is performed in a double-integrator system. Considering the same number of degrees of freedom, the attraction domain of the proposed MPC is equivalent to the one produced by the controller of Limon *et al.* (2008). Moreover, results have shown the closed-loop trajectories obtained with these controllers are the same when there is no model error or disturbances. However, while the proposed strategy has a built-in integral action, the MPC of Limon *et al.* (2008) needs to augment the system with disturbance estimates to remove offset in the controlled variables in the presence of plant-model mismatch and constant unmeasured disturbances. Results have also shown that, when input increments are constrained in the proposed MPC with OPOM, the previous input applied to the system affects the set of feasible initial states. Moreover, although this type of constraint may slow down system convergence to the setpoint, it can be an essential feature in some practical applications.

First formulated for the setpoint tracking problem, the proposed MPC was then extended to address the case of output control zones and optimizing input targets, in which steady-state control objectives are posed within an offset cost function. It is shown that, once the control problem is feasible, it remains feasible despite any changes in control zones or input targets. The system is steered to an output setpoint inside the control zone and to the input target if this operating point is admissible. If not, the system is steered to an operating point that minimizes the offset cost function. The proposed dual-mode MPC with OPOM for handling output zone control and input targets was applied to an unstable reactor system with changing input targets.

Chapter 4

Robust Cooperative Distributed MPC: a multi-model approach

4.1 Introduction

Despite the development of nonlinear MPC strategies, the application of MPC with linear models is still dominant in industry. The reason for this is that methods for the identification of linear models are easy to implement and also because, under mild nonlinearities, a linear model may be a reasonable representation of plant dynamics around an operating point. However, in order to satisfy different economic criteria, chemical plants usually work with changing operating points, which may lead to performance degradation of an MPC controller or even instabilities when the system is far from the operating point in which the linear model was obtained.

A convenient and simple way to model this type of uncertainty is by considering a finite family of linear models that satisfactorily represent the plant at different operating points. This strategy has been successfully applied to a multi-model MPC for controlling an industrial distillation column (C3/C4 splitter) (PORFÍRIO; ALMEIDA NETO; ODLOAK, 2003). Also, some commercial products for identifying linear models that support the multi-model strategy have been reported in a survey on industrial MPC applications (QIN; BADGWELL, 2003). From a theoretical perspective, a stabilizing robust MPC that considers this multi-model uncertainty was first proposed by Badgwell (1997) for open-loop stable systems. This idea has been extended and applied to several robust MPC approaches, e.g. considering the setpoint-tracking problem for stable systems (RODRIGUES; ODLOAK, 2003b), for unstable systems with time delay (MARTINS; ODLOAK, 2016) and for changing economic criterion (FERRAMOSCA; GONZALEZ; LIMON, 2017).

Chemical plants can be viewed as a collection of subsystems that are interconnected and interact with each other through material and energy streams. Since decentralized MPC structures do not account for this interactive nature of chemical processes, they usually have poor performance in comparison to a centralized MPC strategy that, on the other hand, may be difficult to maintain and organize in the case of large-scale systems. For instance, a common maintenance task in MPC controllers is the model update, which may be cumbersome in a single-model centralized MPC, not to mention in a multi-model one. Besides, subsystems have distinct levels of nonlinearity and a different number of possible operating points, thereby requiring a diverse number of linear models to be used in the implementation of a multi-model MPC.

As an alternative framework to perform plantwide control, distributed MPC (DMPC) algorithms provide organizational flexibility and require little effort of maintainability, allowing for gradual commissioning and local reconfiguration. Also, unlike a decentralized control structure, DMPC takes into account the interactions between subsystems and an iterative procedure may be used to improve the overall solution. In general, DMPC strategies differ in the amount of information the subsystems share with each other and the objective they pursue. In noncooperative DMPC approaches, each subsystem optimizes a local performance index and the plantwide system is driven to a Nash equilibrium. Alternatively, in cooperative DMPC strategies all the agents share the same performance criterion, which converges to a Pareto optimum (SCATTOLINI, 2009).

A cooperative DMPC that retains closed-loop stability if the iterative procedure is terminated prior to convergence has been proposed in Stewart *et al.* (2010). When iterating to convergence, this strategy produces the same solution as the one of a centralized MPC scheme, corresponding to the plantwide Pareto optimum. This idea was extended in Ferramosca *et al.* (2013) by using artificial references and a suitable offset cost function, ensuring feasibility under any setpoint change. Separable terminal costs and time-varying local terminal sets were used as stability ingredients in the cooperative DMPC proposed in Conte *et al.* (2016). Another strategy suitable for setpoint tracking has been proposed by Razzanelli and Pannocchia (2017), in which the graph theory was applied to reduce the dimension of dynamic prediction and steady-state models, thereby resulting in a DMPC algorithm with lower computational and communication requirements, while retaining centralized optimality of cooperative schemes.

Recently, cooperative distributed MPC algorithms based on OPOM have also been proposed. In Santana, Martins and Odloak (2020b), a cooperative DMPC with soft terminal constraints is proposed and appropriate conditions on weighting control increments are provided to guarantee closed-loop stability. As an extension of this approach, Sarapka, Martins and Odloak (2021) propose a cooperative DMPC with control zones and input targets, which has no tuning requirements regarding the input move suppression weights for guaranteeing stability. Further details of these works can be found in Santana (2020) and in Sarapka (2020).

Concerning the literature on robust DMPC, different works have proposed solutions for dealing with persistent disturbances and model uncertainty. Considering bounded disturbances, a distributed closed-loop operation of existing centralized robust MPCs is proposed by Conte *et al.* (2013) and a noncooperative robust DMPC strategy based on the constraint tightening approach is presented in Richards and How (2007). In Shalmani, Rahmani and Bigdeli (2020), a Nash-based robust DMPC is proposed for large-scale systems with polytopic uncertainties. Hereafter, we will here focus only on robust DMPC approaches that are based on cooperative algorithms. An LMI-based robust DMPC suitable for cooperative, noncooperative and decentralized control objectives was proposed by Al-Gherwi, Budman and Elkamel (2011) for the case of polytopic uncertainty. This method was improved in Al-Gherwi, Budman and Elkamel (2013) by considering a dual-mode approach, reducing online computations compared to the algorithm previously proposed by the authors in Al-Gherwi, Budman and Elkamel (2011). In Trodden and Richards (2013), a robust DMPC algorithm is proposed in which local agents make hypothetical plans for other agents while optimizing the weighted costs of all subsystems, thereby resulting in cooperation. Under bounded disturbances, problem feasibility is guaranteed by letting only one agent optimize per time step and maintaining previous solutions for the other subsystems. A robustness constraint was employed in the robust cooperative dual-mode DMPC presented in Li and Shi (2014) for nonlinear systems with external bounded disturbances. Using noncooperative distributed moving horizon estimation, an output-feedback robust cooperative DMPC is proposed by Razavinasab, Farsangi and Barkhordari (2018) for linear systems with bounded disturbances and time-varying communication delays. More recently, Wang and Manzie (2020) proposed a robust cooperative DMPC algorithm based on a set-membership constraint tightening approach, avoiding the use of robust positively invariant sets. In the hierarchical robust coalitional MPC presented in Masero *et al.* (2021), an upper layer computes optimal new topology and transition time that are used by the agents in a lower layer to adapt their predicted trajectories while solving their control problem.

Aiming for a more tractable and implementable algorithm, the multi-model approach already employed in robust MPC formulations with OPOM (ODLOAK, 2004; MARTINS; ODLOAK, 2016) is used here to propose a robust cooperative DMPC. The resulting strategy addresses the case in which distributed systems are subject to model uncertainties that arise due to changing operating conditions. The distributed algorithm is built upon the cooperative scheme presented in Stewart *et al.* (2010) and has guaranteed recursive feasibility, convergence and stability. Based on the OPOM model, the proposed robust DMPC produces offset-free control and is suitable for both setpoint tracking and the so-called zone control strategy. Moreover, it can be seen as an extension of recently published cooperative DMPC formulations based on OPOM (SANTANA; MARTINS; ODLOAK, 2020b; SARAPKA; MARTINS; ODLOAK, 2021).

Based on the approach presented here, two alternative algorithms are proposed. One improves convergence over iterations by computing weights of the convex combination of agents' solutions such that the plantwide performance index is minimized. The other turns the optimization problem solved by each agent into a QP (quadratic programming) problem by enforcing the robustness constraint only after iterations terminate. This strategy reduces the number of QCQP (quadratically constrained quadratic programming) problems solved at each time step, which also reduces the CPU time spent by the agents. The proposed methods are applied to two case studies and compared to the centralized robust MPC. The first two algorithms proposed in this chapter have been published in Sencio and Odloak (2022).

This chapter is organized as follows. Section 4.2 presents preliminary definitions that involves the considered system model, constraints, performance index as well as the multi-model uncertainty. In Section 4.3, the robust cooperative distributed MPC (Algorithm 1) is formulated and recursive feasibility, convergence and stability analyses are presented. Numerical simulations of the proposed Algorithm 1 applied to a high-purity distillation column are presented in Section 4.4. Section 4.5 provides an alternative approach for improving algorithm convergence (Algorithm 2) and a method for robustifying a nominal DMPC (Algorithm 3) is presented in Section 4.6 along with theoretical analyses. The proposed algorithms are applied to a two reactors-separator process in Section 4.7. Finally, Section 4.8 presents some concluding remarks.

4.2 Preliminaries

4.2.1 Velocity-form model

Consider the time-invariant, discrete-time state-space model in the positional form of inputs defined below, in which $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$ and n_x, n_u and n_y are the number of states, inputs and outputs, respectively:

$$x^+ = Ax + Bu \tag{4.1}$$

$$y = Cx \tag{4.2}$$

Assumption 4. Matrix A is discrete-time Hurwitz.

As shown in Chapter 2, the following OPOM-like model can be obtained:

$$\begin{bmatrix} y_s \\ x_d \end{bmatrix}^+ = \begin{bmatrix} I_{n_y} & 0 \\ 0 & F \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} + \begin{bmatrix} B_s \\ B_d \end{bmatrix} \Delta u \tag{4.3}$$

$$y = \begin{bmatrix} I_{n_y} & \Psi \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} \tag{4.4}$$

in which

$$F = A, \qquad B_s = C\left(I - A\right)^{-1} B, \qquad B_d = -A\left(I - A\right)^{-1} B, \qquad \Psi = C.$$

In the model given in (4.3) and (4.4), y_s is an artificial integrating state that appears when writing the model in the incremental form of inputs and corresponds to the prediction of system steady output (static part); and x_d is related to the stable dynamic part of the original system (4.1)-(4.2).

4.2.2 Subsystems and plantwide model

The centralized model (4.1)-(4.2) can be partitioned into M subsystems coupled through the inputs (see (STEWART *et al.*, 2010), Appendix B) such that each one is represented by a (possibly non-minimal) state-space model of the following form:

$$\begin{aligned} x_i^+ &= A_i x_i + B_{ii} u_i + \sum_{\substack{j=1\\ j\neq i}}^{M} B_{ij} u_j, \quad && i \in \mathbb{I}_{1:M} \\ y_i &= C_i x_i, && i \in \mathbb{I}_{1:M} \end{aligned}$$

In the above model, subsystem i has state $x_i \in \mathbb{R}^{n_{x_i}}$, input $u_i \in \mathbb{R}^{n_{u_i}}$, output $y_i \in \mathbb{R}^{n_{y_i}}$ and matrices $A_i \in \mathbb{R}^{n_{x_i}\times n_{x_i}}$, $B_{ii} \in \mathbb{R}^{n_{x_i}\times n_{u_i}}$, $B_{ij} \in \mathbb{R}^{n_{x_i}\times n_{u_j}}$ and $C_i \in \mathbb{R}^{n_{y_i}\times n_{x_i}}$. Note that the input of every subsystem $j \neq i$ can be viewed as a disturbance to subsystem i. The plantwide representation of a collection of M subsystems is:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}^+ = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_M \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} + \sum_{i=1}^{M} \begin{bmatrix} B_{1i} \\ B_{2i} \\ \vdots \\ B_{Mi} \end{bmatrix} u_i$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_M \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}$$

which can be represented with a little abuse of notation in a simplified form as follows:

$$x^+ = Ax + \sum_{i=1}^{M} B_i u_i \tag{4.5}$$

$$y = Cx \tag{4.6}$$

in which we redefine $n_x = \sum_{i=1}^{M} n_{x_i}$ and $x \in \mathbb{R}^{n_x}$, $u_i \in \mathbb{R}^{n_{u_i}}$, $y \in \mathbb{R}^{n_y}$, $A \in \mathbb{R}^{n_x\times n_x}$, $B_i \in \mathbb{R}^{n_x\times n_{u_i}}$ and $C \in \mathbb{R}^{n_y\times n_x}$.

Using the proposed method to obtain a model in the velocity-form of inputs, system (4.5)-(4.6) can be split into static and dynamic parts as follows:

$$\begin{bmatrix} y_s \\ x_d \end{bmatrix}^+ = \begin{bmatrix} I_{n_y} & 0 \\ 0 & F \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} + \sum_{i=1}^{M} \begin{bmatrix} B_{s,i} \\ B_{d,i} \end{bmatrix} \Delta u_i \tag{4.7}$$

$$y = \begin{bmatrix} I_{n_y} & \Psi \end{bmatrix} \begin{bmatrix} y_s \\ x_d \end{bmatrix} \tag{4.8}$$

in which

$$F = A, \quad B_{s,i} = C\left(I_{n_x} - F\right)^{-1} B_i, \quad B_{d,i} = -F\left(I_{n_x} - F\right)^{-1} B_i, \quad \Psi = C.$$

Throughout this chapter, we use a redefined notation given as follows:

$$x \leftarrow \begin{bmatrix} y_s \\ x_d \end{bmatrix}, \quad A \leftarrow \begin{bmatrix} I_{n_y} & 0 \\ 0 & F \end{bmatrix}, \quad B_i \leftarrow \begin{bmatrix} B_{s,i} \\ B_{d,i} \end{bmatrix}, \quad C \leftarrow \begin{bmatrix} I_{n_y} & \Psi \end{bmatrix}.$$

4.2.3 System constraints

Here, we consider that the system is subject to linear constraints on inputs and input movements. For this purpose, the following sets can be defined:

$$\mathcal{U} := \{u = (u_1, \dots, u_M) \in \mathbb{R}^{n_u} : A_{u,i} u_i \leq b_{u,i},\ i \in \mathbb{I}_{1:M}\} \tag{4.9}$$

$$\mathcal{U}_\Delta := \{\Delta u = (\Delta u_1, \dots, \Delta u_M) \in \mathbb{R}^{n_u} : A_{\Delta,i} \Delta u_i \leq b_{\Delta,i},\ i \in \mathbb{I}_{1:M}\} \tag{4.10}$$

Since we will deal with the problem of zone control, the plantwide output setpoint y_{sp} is assumed to be in a given zone or range defined by the following set:

$$\mathcal{Y}_{sp} := \{y_{sp} \in \mathbb{R}^{n_y} : y_{sp,min} \leq y_{sp} \leq y_{sp,max}\} \tag{4.11}$$

As already mentioned in Chapter 3, the setpoint tracking problem is a particular case of the zone control strategy in which $y_{sp} = y_{sp,min} = y_{sp,max}$.

4.2.4 The plantwide performance index

In a cooperative game, all the agents share the same objective. This means that, concerning a cooperative distributed MPC framework, each agent i computes a sequence of control movements $\Delta \mathbf{u}_i = \{\Delta u_i(0), \dots, \Delta u_i(N-1)\}$ such that it minimizes a plantwide performance index and satisfies the constraints of all subsystems. Let $y_{sp} \in \mathbb{R}^{n_y}$ define a given output setpoint, then the state setpoint for system (4.7) is simply given by $x_{sp} = (y_{sp}, 0) \in \mathbb{R}^{n_y + n_x}$. Now, consider the plantwide performance criterion $V(x, \Delta \mathbf{u}, y_{sp})$ given as follows:

$$V(x, \Delta \mathbf{u}, y_{sp}) = \sum_{j=0}^{\infty} \|x(j) - x_{sp}\|_Q^2 + \|\Delta u(j)\|_R^2 \tag{4.12}$$

in which $Q \in \mathbb{R}^{(n_y+n_x)\times(n_y+n_x)}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are positive definite weighting matrices. To simplify our developments, we consider $Q = \left[\begin{smallmatrix} Q_s & Q_{sd} \\ Q_{sd}^T & Q_d \end{smallmatrix}\right]$, with $Q_s \in \mathbb{R}^{n_y \times n_y}$, $Q_d \in \mathbb{R}^{n_x \times n_x}$ and $Q_{sd} \in \mathbb{R}^{n_y \times n_x}$. The states $x(j) \in \mathbb{R}^{n_y+n_x}$ are predicted through model (4.7) with $x(0) = x$ and control movements $\Delta u(j)$ that are elements of the sequence $\Delta\mathbf{u}$.

Remark 4. As will be clearer later on, the chosen structure of matrix Q is useful to write the penalization of y_s and x_d separately. Moreover, Q_{sd} represents a nonzero submatrix of Q that appears when choosing $Q = C^T Q_y C$, with $Q_y \in \mathbb{R}^{n_y \times n_y}$ positive definite, which is typically used to penalize the predicted output errors with respect to y_{sp}.

Beyond a control horizon N, we consider here the zero control policy in which $\Delta u(N + j) = 0$ for $j \in \mathbb{I}_{\geq 0}$. The infinite sum in (4.12) can be divided into two parts as follows:

$$\begin{aligned}\sum_{j=0}^{\infty} \|x(j) - x_{sp}\|_Q^2 + \|\Delta u(j)\|_R^2 &= \sum_{j=0}^{N-1} \|x(j) - x_{sp}\|_Q^2 + \|\Delta u(j)\|_R^2 + \sum_{j=N}^{\infty} \|x(j) - x_{sp}\|_Q^2 \\ &= \sum_{j=0}^{N-1} \|x(j) - x_{sp}\|_Q^2 + \|\Delta u(j)\|_R^2 + \sum_{j=0}^{\infty} \Big(\|y_s(N+j) - y_{sp}\|_{Q_s}^2 \\ &\quad + \|x_d(N+j)\|_{Q_d}^2 + 2\left(y_s(N+j) - y_{sp}\right)^T Q_{sd} x_d(N+j) \Big)\end{aligned}$$

From (4.7), we obtain $y_s(N + j)$ and $x_d(N + j)$, resulting in $y_s(N + j) = y_s(N)$ and $x_d(N + j) = F^j x_d(N)$ for $j \in \mathbb{I}_{\geq 0}$. In order to prevent the control cost from being unbounded, we must ensure that $y_s(N) - y_{sp} = 0$, then we have that:

$$\begin{aligned}\sum_{j=0}^{\infty} \Big(\|y_s(N+j) - y_{sp}\|_{Q_s}^2 + \|x_d(N+j)\|_{Q_d}^2 + 2\left(y_s(N+j) - y_{sp}\right)^T Q_{sd} x_d(N+j) \Big) \\ = \sum_{j=0}^{\infty} \|F^j x_d(N)\|_{Q_d}^2 \\ = x_d^T(N) \left(\sum_{j=0}^{\infty} \left(F^j\right)^T Q_d F^j \right) x_d(N) \\ = x_d^T(N) P x_d(N)\end{aligned}$$

Since F is stable, the infinite sum P converges and can thus be computed. For this purpose, multiply P by F^T from the left and by F from the right and observe that:

$$P - F^T P F = \sum_{j=0}^{\infty} \left(F^j\right)^T Q_d F^j - \sum_{j=1}^{\infty} \left(F^j\right)^T Q_d F^j$$

which results in the following Lyapunov equation:

$$P - F^T P F = Q_d \tag{4.13}$$

Then, the control cost (4.12) can be rewritten as follows:

$$V(x, \Delta\mathbf{u}, y_{sp}) = \sum_{j=0}^{N-1} \left(\|x(j) - x_{sp}\|_Q^2 + \|\Delta u(j)\|_R^2 \right) + \|x_d(N)\|_P^2 \tag{4.14}$$

However, in a constrained system, the terminal constraint $y_s(N) - y_{sp} = 0$ may not be satisfied due to conflicts with input constraints (4.9) and (4.10), which leads to infeasibilities. Following the approach of Rodrigues and Odloak (2003b), the terminal constraint is softened as $y_s(N) - y_{sp} - \delta = 0$, in which $\delta \in \mathbb{R}^{n_y}$ is an additional degree of freedom that corresponds to the offset between the terminal predicted steady output $y_s(N)$ and the output setpoint y_{sp}. This relaxation is then penalized in the control cost, which can be redefined as follows:

$$V(x, \Delta \mathbf{u}, y_{sp}, \delta) = \sum_{j=0}^{N-1} \left(\|x(j) - x_{sp} - \delta_x\|_Q^2 + \|\Delta u(j)\|_R^2 \right) + \|x_d(N)\|_P^2 + \|\delta\|_S^2 \tag{4.15}$$

in which $\delta_x = (\delta, 0) \in \mathbb{R}^{n_y+n_x}$ and $S \in \mathbb{R}^{n_y \times n_y}$ is a positive definite weighting matrix.

Let x_s be the artificial reference defined as follows:

$$x_s = x_{sp} + \delta_x = \begin{bmatrix} y_{sp} + \delta \\ 0 \end{bmatrix} = \begin{bmatrix} y_s(N) \\ 0 \end{bmatrix} \tag{4.16}$$

Then, by substituting $\delta = y_s(N) - y_{sp}$, the following plantwide performance criterion is obtained:

$$V(x, \Delta \mathbf{u}, y_{sp}) = \sum_{j=0}^{N-1} \left(\|x(j) - x_s\|_Q^2 + \|\Delta u(j)\|_R^2 \right) + \|x_d(N)\|_P^2 + \|y_s(N) - y_{sp}\|_S^2 \tag{4.17}$$

Remark 5. As also pointed out in Chapter 3, that the last term in (4.17) corresponds to the penalization of the slack variable δ employed in Rodrigues and Odloak (2003b) which is equivalent to the offset cost function proposed by Limon *et al.* (2008).

Remark 6. In a cooperative approach, the agents have the same objective and thus need to predict the plantwide output, even if they only manipulate their local inputs. This is why the performance index employed here is defined in terms of the overall system variables, which has also been used in Ferramosca *et al.* (2013). Equivalently, a performance criterion can also be defined in terms of the weighted sum of objective functions written individually for each subsystem (STEWART *et al.*, 2010), which is employed in other DMPC formulations based on OPOM (SANTANA; MARTINS; ODLOAK, 2020b; SARAPKA; MARTINS; ODLOAK, 2021). However, compared with Santana, Martins and Odloak (2020b) and Sarapka, Martins and Odloak (2021), the approach used in this work results in a more compact performance index. Moreover, unlike in Sarapka, Martins and Odloak (2021), the use of a large set of slack variables related to the offset in the outputs is avoided here, which simplifies the control problem.

4.2.5 Multi-model uncertainty

Model uncertainty arises when system matrices A, B and C are not exactly known. In order to account for different operating conditions, one can consider a finite set of L

models $\theta = (A_\theta, B_\theta, C_\theta)$ defined as $\Omega := \{\theta_1, \dots, \theta_L\}$ and select a model as the one that most likely represent the plant, referred to as the nominal model $\theta_n \in \Omega$. Let us assume that the true plant model θ_t is such that $\theta_t \in \Omega$, although we do not know which model of set Ω is the true one (BADGWELL, 1997).

Using the plantwide representation of the partitioned system given in (4.7) and (4.8) and considering the presence of multi-model uncertainty, one can define models $\theta = (F_\theta, B_{s,\theta,i}, B_{d,\theta,i}, \Psi_\theta, i \in \mathbb{I}_{1:M}) = (A_\theta, B_{\theta,i}, C_\theta, i \in \mathbb{I}_{1:M})$ such that $\theta \in \Omega$. In this case, the control cost (4.17) can be written for each plantwide model $\theta \in \Omega$ as follows:

$$V_\theta(x_\theta, \Delta\mathbf{u}, y_{sp}) = \sum_{j=0}^{N-1} \left(\|x_\theta(j) - x_{s,\theta}\|_Q^2 + \|\Delta u(j)\|_R^2 \right) + \|x_{d,\theta}(N)\|_{P_\theta}^2 + \|y_{s,\theta}(N) - y_{sp}\|_S^2 \tag{4.18}$$

in which subscript θ denotes the dependency on model θ and P_θ is computed through the following Lyapunov equation written for a given model θ:

$$P_\theta - F_\theta^T P_\theta F_\theta = Q_d, \quad \theta \in \Omega \tag{4.19}$$

Assumption 5. All models $\theta \in \Omega$ have gain matrices with the same sign of input/output directions (SKOGESTAD; POSTLETHWAITE, 2005).

4.3 A robust cooperative distributed MPC

4.3.1 Problem formulation

Assume that a feasible initial solution $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ is available. Then, at each iteration $p \in \mathbb{I}_{\geq 1}$, each agent $i \in \mathbb{I}_{1:M}$ solves the following control problem denoted as $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$:

$$(\Delta\mathbf{u}_i^*, y_{sp,i}^*) = \arg \min_{\Delta\mathbf{u}_i, y_{sp,i}} \quad V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}, y_{sp}\right) \tag{4.20}$$

subject to:

$$x_\theta(0) = x_{\theta_t}, \qquad \theta \in \Omega \tag{4.21}$$

$$x_\theta(j+1) = A_\theta x_\theta(j) + \sum_{i=1}^{M} B_{\theta,i} \Delta u_i(j), \quad j \in \mathbb{I}_{0:N-1},\ \theta \in \Omega \tag{4.22}$$

$$\Delta\mathbf{u}_l = \Delta\mathbf{u}_l^{[p-1]}, \qquad l \in \mathbb{I}_{1:M} \setminus i \tag{4.23}$$

$$u(j) \in \mathcal{U}, \qquad j \in \mathbb{I}_{0:N-1} \tag{4.24}$$

$$\Delta u(j) \in \mathcal{U}_\Delta, \qquad j \in \mathbb{I}_{0:N-1} \tag{4.25}$$

$$y_{sp} \in \mathcal{Y}_{sp} \tag{4.26}$$

$$V_\theta(x_\theta, \Delta\mathbf{u}, y_{sp}) \leq V_\theta(x_\theta, \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}), \qquad \theta \in \Omega \tag{4.27}$$

in which $\Delta\mathbf{u}_i^*$ is the optimal control movement of subsystem i, $y_{sp,i}^*$ is the optimal plantwide output setpoint computed by agent i and (4.27) corresponds to a robustness constraint that enforces the performance criterion of each plantwide model $\theta \in \Omega$ to be no greater than the control cost evaluated with the feasible initial solution.

Considering the positiveness of weighting matrices, the performance index (4.18) used as the objective function of problem $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ is, by construction, a strictly convex function of the decision variables, which implies that constraint (4.27) is convex. Therefore, since $\mathcal{U}$, $\mathcal{U}_\Delta$ and $\mathcal{Y}_{sp}$ are convex sets, problem $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ is a convex optimization problem and, consequently, every local minimum is a global minimum. Moreover, since the objective function is strictly convex, this implies that the solution, if exists, is unique. In fact, given that the objective function and the robustness constraint (4.27) are quadratic and the remaining constraints are linear, the proposed control problem consists of a convex quadratically constrained quadratic program (QCQP), which can be solved via nonlinear programming (NLP). A convex QCQP can also be cast as a second-order cone program (SOCP), which, in turn, can be expressed as a semi-definite program (SDP) by writing constraints as a linear matrix inequality. However, solving SOCPs via NLP or SDP is not recommended since specialized interior-point methods that solve SOCPs directly are computationally more efficient (LOBO *et al.*, 1998).

Remark 7. The proposed robust cooperative distributed MPC can be extended for dealing with integrating and unstable systems. In this case, the control problem $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ must also include terminal constraints related to integrating and unstable modes of the system. As an alternative to improve problem feasibility, these terminal constraints can be softened by using slack variables, as proposed in Santoro and Odloak (2012) and Martins and Odloak (2016).

After each agent i has solved its respective problem $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$, the solutions are combined to produce a global solution $(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ as follows:

$$(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}) = \sum_{i=1}^{M} w_i(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*) \tag{4.28}$$

in which

$$\begin{gathered}\Delta\bar{\mathbf{u}}_i^* = (\Delta\mathbf{u}_1^{[p-1]}, \ldots, \Delta\mathbf{u}_i^*, \ldots, \Delta\mathbf{u}_M^{[p-1]}) \\ \sum_{i=1}^{M} w_i = 1, \quad w_i > 0, \forall i \in \mathbb{I}_{1:M}\end{gathered}$$

Once the solutions of each agent are combined to produce $(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$, one can move to the next iteration $p+1$ or stop if the maximum number of iterations has been achieved or if the current solution meets some accuracy specification. When the iterative procedure terminates at iteration $\bar{p}$ at a given time step k, the best available solution is

$(\Delta\mathbf{u}_k, y_{sp,k}) = (\Delta\mathbf{u}_k^{[\bar{p}]}, y_{sp,k}^{[\bar{p}]})$ which is then used to build an initial condition for the first iteration to be performed at time step $k+1$ as follows:

$$\Delta\mathbf{u}_{k+1}^{[0]} = \{\Delta u(1|k), \ldots, \Delta u(N-1|k), 0\} \tag{4.29}$$

$$y_{sp,k+1}^{[0]} = y_{sp,k} \tag{4.30}$$

By following the receding horizon principle, the first control movement of $\Delta\mathbf{u}_k$ is applied to the system, i.e. $\Delta u(k) = \Delta u(0|k)$, and this procedure is then repeated at any subsequent time step $k+j > k$, which implicitly generates the MPC control law denoted as $\kappa_N(x_{\theta_t}, \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$.

The proposed robust cooperative distributed MPC with multi-model uncertainty can be summarized in the following algorithm:

Algorithm 1: Robust cooperative distributed MPC

```
   Input: $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ and $\mathcal{Y}_{sp}$
 1 Set $x = x_{\theta_t}(k)$ and $p = 1$;
 2 while stop criteria are not met do
 3     for $i \in \mathbb{I}_{1:M}$ do
 4         Obtain $(\Delta\mathbf{u}_i^*, y_{sp,i}^*)$ by solving $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$;
 5     end
 6     Compute $(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ through (4.28);
 7     $p \leftarrow p + 1$;
 8 end
 9 Set $(\Delta\mathbf{u}_k, y_{sp,k}) = (\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$;
10 Compute $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ through (4.29) and (4.30);
11 Implement $\Delta u(0|k)$;
12 $k \leftarrow k + 1$;
13 Return to line 1.
```

Note that, for a given control zone $\mathcal{Y}_{sp}$, the above algorithm can always be initialized by choosing $\Delta\mathbf{u}^{[0]} = \{0, \ldots, 0\}$ and any $y_{sp}^{[0]} \in \mathcal{Y}_{sp}$.

4.3.2 Recursive feasibility, convergence and stability analyses

The recursive feasibility of problem $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ of an iteration p is addressed as follows:

Lemma 5 (Recursive feasibility over iterations)**.** *Assume that the initial condition $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ is feasible and consider the application of Algorithm 1. Then, problem $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ is feasible at iteration $p = 1$ for any agent $i \in \mathbb{I}_{1:M}$ and it will remain feasible at any subsequent iteration $p + j,\ j \in \mathbb{I}_{\geq 1}$.*

Proof. First, since we have assumed that a feasible initial condition $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ is available, it is easy to see that a feasible solution to $P_i(x; \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ is the initial condition itself and, consequently, every agent $i \in \mathbb{I}_{1:M}$ will have a solution at $p = 1$. A convex combination of these solutions leads to $(\Delta\mathbf{u}^{[1]}, y_{sp}^{[1]})$, which is used as an initial condition for iteration $p = 2$ and clearly satisfies constrains (4.24)-(4.26) because $\mathcal{U}$, $\mathcal{U}_\Delta$ and $\mathcal{Y}_{sp}$ are convex sets. Since constraint (4.27) is satisfied for each subsystem $i \in \mathbb{I}_{1:M}$ at $p = 1$, then we can state that:

$$\sum_{i=1}^{M} w_i V_\theta\left(x_\theta, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right) \leq \sum_{i=1}^{M} w_i V_\theta\left(x_\theta, \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}\right) = V_\theta\left(x_\theta, \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}\right), \quad \forall\theta \in \Omega$$

Then, from the convexity of the control cost, it follows that:

$$V_\theta(x_\theta, \Delta\mathbf{u}^{[1]}, y_{sp}^{[1]}) \leq \sum_{i=1}^{M} w_i V_\theta\left(x_\theta, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right) \leq V_\theta\left(x_\theta, \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}\right), \quad \forall\theta \in \Omega$$

which implies that $(\Delta\mathbf{u}^{[1]}, y_{sp}^{[1]})$ satisfies (4.27) and, thus, serves as a feasible solution to all agents $i \in \mathbb{I}_{1:M}$.

Therefore, by induction, we conclude that $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ will be feasible for all agents $i \in \mathbb{I}_{1:M}$ at any subsequent iteration $p + j,\ j \in \mathbb{I}_{\geq 1}$. □

Corollary 2 (Recursive feasibility over time steps)**.** *Consider the application of Algorithm 1. The feasibility of $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ at iteration $p = 1$ implies the feasibility at every time step $k + j > k$.*

Proof. This is a direct result from Lemma 5 and the construction of a feasible initial condition $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ given in (4.29) and (4.30). □

Since we are minimizing the cost computed for the nominal model $\theta_n \in \Omega$, we expect it to be non-increasing over iterations and to converge, which is ensured by the following lemma:

Lemma 6 (Convergence of the algorithm)**.** *In the application of Algorithm 1, the plantwide performance criterion $V_{\theta_n}(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ of the nominal model $\theta_n \in \Omega$ is non-increasing and converges as $p \to \infty$.*

Proof. After solving $P_i(x; \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ at an iteration $p + 1$, each agent $i \in \mathbb{I}_{1:M}$ obtains a value function such that $V_{\theta_n}\left(x_{\theta_n}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right) \leq V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}\right)$. A convex combination of these value functions satisfies the following relationship:

$$\sum_{i=1}^{M} w_i V_{\theta_n}\left(x_{\theta_n}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right) \leq \sum_{i=1}^{M} w_i V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}\right) = V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}\right)$$

Therefore, by the convexity of the control cost, we have that:

$$V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p+1]}, y_{sp}^{[p+1]}\right) \leq \sum_{i=1}^{M} w_i V_{\theta_n}\left(x_{\theta_n}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right) \leq V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}\right)$$

This shows that the control cost is non-increasing over iterations. Since the control cost is non-negative by construction, it is bounded below by zero and, therefore, converges. □

Lemma 7 (Auxiliary results)**.** *Consider an admissible output setpoint $y_{sp} \in \mathcal{Y}_{sp}$ and a given initial state $x = (y_s, 0)$ with $y_s \neq y_{sp}$ and a predecessor input $u^- = u_s \in int(\mathcal{U})$. Then, there exists an admissible control movement Δu that steers y_s to $y^+_{s,\theta}$ such that the following relationship holds for a positive definite matrix S:*

$$\|y_s - y^+_{s,\theta}\|^2_{Q_s} + \|\Delta u\|^2_R + \|x^+_{d,\theta}\|^2_{P_\theta} + \|y^+_{s,\theta} - y_{sp}\|^2_S < \|y_s - y_{sp}\|^2_S, \ \forall \theta \in \Omega$$

in which $y^+_{s,\theta} = y_s + B_{s,\theta}\Delta u$ and $x^+_{d,\theta} = B_{d,\theta}\Delta u$.

Proof. Consider a steady output given as $y^+_{s,\theta} = \alpha_\theta y_s + (1-\alpha_\theta) y_{sp}$ with $\alpha_\theta \in (0,1), \ \forall\theta \in \Omega$ and assume $B_{s,\theta}, \ \forall\theta\in\Omega$, has full rank. From (4.7), the control movement Δu_θ needed to steer y_s to $y^+_{s,\theta}$ is such that $y^+_{s,\theta} = y_s + B_{s,\theta}\Delta u_\theta, \ \forall\theta\in\Omega$ and, from the definition of $y^+_{s,\theta}$, we have that $y^+_{s,\theta} - y_s = (1-\alpha_\theta)(y_{sp} - y_s)$. Then, since $u_s \in \text{int}(\mathcal{U})$, there exists a sufficiently large $\alpha_\theta \in (0,1)$ such that $(1-\alpha_\theta)(y_{sp} - y_s) = B_{s,\theta}\Delta u_\theta$ holds for all models $\theta \in \Omega$ with $\Delta u_\theta \in \mathcal{U}_\Delta$ and $(u_s + \Delta u_\theta) \in \mathcal{U}$, which means that every model has an admissible control movement that steers the predicted steady output closer to the output setpoint.

Now, observe that steering y_s to $y^+_{s,\theta}$ with a control sequence $\Delta\mathbf{u}_\theta = \{\Delta u_\theta, 0, \dots, 0\}$ results in the following control cost:

$$\begin{aligned} V_\theta(x, \Delta\mathbf{u}_\theta, y_{sp}) &= \sum_{j=0}^{N-1}\left(\|x_\theta(j) - x_{s,\theta}\|^2_Q\right) + \|\Delta u_\theta\|^2_R + \|x_{d,\theta}(N)\|^2_{P_\theta} + \|y_{s,\theta}(N) - y_{sp}\|^2_S \\ &= \|y_s - y^+_{s,\theta}\|^2_{Q_s} + \|x^+_{d,\theta}\|^2_{Q_d} + \|F_\theta x^+_{d,\theta}\|^2_{Q_d} + \cdots + \|F_\theta^{N-2} x^+_{d,\theta}\|^2_{Q_d} \\ &\quad + \|\Delta u_\theta\|^2_R + \|F_\theta^{N-1} x^+_{d,\theta}\|^2_{P_\theta} + \|y^+_{s,\theta} - y_{sp}\|^2_S, \quad \theta\in\Omega \end{aligned}$$

in which $x_{s,\theta} = (y_{s,\theta}(N), 0)$, $y_{s,\theta}(N) = y^+_{s,\theta} = y_s + B_{s,\theta}\Delta u_\theta$ and $x_{d,\theta}(N) = F_\theta^{N-1}x^+_{d,\theta}$, with $x^+_{d,\theta} = B_{d,\theta}\Delta u_\theta$. Noting that $P_\theta = \sum_{j=0}^{\infty}\left(F_\theta^j\right)^T Q_d F_\theta^j$, then we have that:

$$\begin{aligned} &\|x^+_{d,\theta}\|^2_{Q_d} + \|F_\theta x^+_{d,\theta}\|^2_{Q_d} + \cdots + \|F_\theta^{N-2}x^+_{d,\theta}\|^2_{Q_d} + \|F_\theta^{N-1}x^+_{d,\theta}\|^2_{P_\theta} \\ &\quad = \left(x^+_{d,\theta}\right)^T\left(Q_d + F_\theta^T Q_d F_\theta + \cdots + \left(F_\theta^{N-2}\right)^T Q_d F_\theta^{N-2} + \sum_{j=N-1}^{\infty}\left(F_\theta^j\right)^T Q_d F_\theta^j\right) x^+_{d,\theta} \\ &\quad = \left(x^+_{d,\theta}\right)^T\left(\sum_{j=0}^{\infty}\left(F_\theta^j\right)^T Q_d F_\theta^j\right) x^+_{d,\theta} = \|x^+_{d,\theta}\|^2_{P_\theta} \end{aligned}$$

Then, the control cost is:

$$V_\theta(x, \Delta\mathbf{u}_\theta, y_{sp}) = \|y_s - y^+_{s,\theta}\|^2_{Q_s} + \|\Delta u_\theta\|^2_R + \|x^+_{d,\theta}\|^2_{P_\theta} + \|y^+_{s,\theta} - y_{sp}\|^2_S, \quad \theta\in\Omega$$

One can derive the following expressions:

$$\|y_s - y^+_{s,\theta}\|^2_{Q_s} = \|\Delta u_\theta\|^2_{\tilde{Q}_{s,\theta}}, \qquad \tilde{Q}_{s,\theta} = B^T_{s,\theta} Q_s B_{s,\theta}$$

$$\begin{aligned}
\|x^+_{d,\theta}\|^2_{P_\theta} &= \|\Delta u_\theta\|^2_{\tilde{P}_\theta}, & \tilde{P}_\theta &= B^T_{d,\theta} P_\theta B_{d,\theta}\\
\|y^+_{s,\theta} - y_{sp}\|^2_S &= \alpha^2_\theta (1-\alpha_\theta)^{-2} \|\Delta u_\theta\|^2_{\tilde{S}_\theta}, & \tilde{S}_\theta &= B^T_{s,\theta} S B_{s,\theta}\\
\|y_s - y_{sp}\|^2_S &= (1-\alpha_\theta)^{-2} \|\Delta u_\theta\|^2_{\tilde{S}_\theta}, & \tilde{S}_\theta &= B^T_{s,\theta} S B_{s,\theta}
\end{aligned}$$

Then, we have that:

$$V_\theta(x, \Delta\mathbf{u}_\theta, y_{sp}) = \|\Delta u_\theta\|^2_{\tilde{Q}_{s,\theta}} + \|\Delta u_\theta\|^2_R + \|\Delta u_\theta\|^2_{\tilde{P}_\theta} + \alpha^2_\theta (1-\alpha_\theta)^{-2} \|\Delta u_\theta\|^2_{\tilde{S}_\theta} = \|\Delta u_\theta\|^2_{H_\theta}$$

in which $H_\theta = \tilde{Q}_{s,\theta} + R + \tilde{P}_\theta + \alpha^2_\theta (1-\alpha_\theta)^{-2} \tilde{S}_\theta$. Now, to prove that $\|y_s - y_{sp}\|^2_S > \|\Delta u_\theta\|^2_{H_\theta}$, which is equivalent to show that $(1-\alpha_\theta)^{-2} \|\Delta u_\theta\|^2_{\tilde{S}_\theta} > \|\Delta u_\theta\|^2_{H_\theta}$, we shall find a sufficiently large α_θ such that $(1-\alpha_\theta)^{-2}\tilde{S}_\theta > H_\theta$. For this purpose, observe that there exists a constant $\varphi > 0$ such that $\varphi \tilde{S}_\theta > \tilde{Q}_{s,\theta} + R + \tilde{P}_\theta$ holds for all $\theta \in \Omega$. Then, we can write the following expression:

$$\begin{aligned}
(1-\alpha_\theta)^{-2}\tilde{S}_\theta - H_\theta &= \left(1-\alpha^2_\theta\right)(1-\alpha_\theta)^{-2}\tilde{S}_\theta - \tilde{Q}_{s,\theta} - R - \tilde{P}_\theta\\
&> \left(1-\alpha^2_\theta\right)(1-\alpha_\theta)^{-2}\tilde{S}_\theta - \varphi\tilde{S}_\theta\\
&= (1-\alpha_\theta)^{-2}\left(1-\alpha^2_\theta - (1-\alpha_\theta)^2\varphi\right)\tilde{S}_\theta\\
&= (1-\alpha_\theta)^{-1}\left(1+\alpha_\theta - (1-\alpha_\theta)\varphi\right)\tilde{S}_\theta
\end{aligned}$$

Therefore, for every $\alpha_\theta \in (\alpha_{min}, 1)$, in which $\alpha_{min} = \max\left(0, \frac{\varphi-1}{\varphi+1}\right)$, we have that $H_\theta < (1-\alpha_\theta)^{-2}\tilde{S}_\theta$ holds for every model $\theta \in \Omega$.

Under Assumption 5, all the models $\theta \in \Omega$ have the same sign of output/input directions. Then, there exists an admissible control movement Δu that produces a steady output given as $y^+_{s,\theta} = y_s + B_{s,\theta}\Delta u$ such that $\alpha_{min}\|y_s - y_{sp}\| < \|y^+_{s,\theta} - y_{sp}\| < \|y_s - y_{sp}\|$, $\forall\theta \in \Omega$, which means that all the models will have a predicted steady output $y^+_{s,\theta}$ closer to the output setpoint y_{sp} and that the control cost satisfies the following relationship:

$$\begin{aligned}
V_\theta(x, \Delta\mathbf{u}, y_{sp}) &= \sum_{j=0}^{N-1}\left(\|x_\theta(j) - x_{s,\theta}\|^2_Q\right) + \|\Delta u\|^2_R + \|x_{d,\theta}(N)\|^2_{P_\theta} + \|y_{s,\theta}(N) - y_{sp}\|^2_S\\
&= \|y_s - y^+_{s,\theta}\|^2_{Q_s} + \|\Delta u\|^2_R + \|x^+_{d,\theta}\|^2_{P_\theta} + \|y^+_{s,\theta} - y_{sp}\|^2_S\\
&< \|y_s - y_{sp}\|^2_S,\ \forall\theta \in \Omega
\end{aligned}$$

in which $x_{s,\theta} = (y_{s,\theta}(N), 0)$, $y_{s,\theta}(N) = y^+_{s,\theta} = y_s + B_{s,\theta}\Delta u$ and $x^+_{d,\theta} = B_{d,\theta}\Delta u$. □

In what follows, $(\Delta\mathbf{u}^*_k, y^*_{sp,k})$ denotes the best available solution at time step k and not an optimal one in the sense of the centralized MPC problem.

Lemma 8 (Convergence of y_s to y_{sp})**.** *Consider an admissible output setpoint $y^*_{sp} \in \mathcal{Y}_{sp}$ and a given initial state $x_\theta = x_{\theta_t}$, $\forall\theta \in \Omega$, with predecessor input $u^- = u_s \in int(\mathcal{U})$. If the combined solutions of all the agents $i \in \mathbb{I}_{1:M}$ computed through* (4.28) *is $(\Delta\mathbf{u}^*, y^*_{sp})$ such that $\|x_\theta - x_{s,\theta}\|^2_Q = 0$, then $\|y_{s,\theta}(N) - y^*_{sp}\|^2_S = 0$, $\forall\theta \in \Omega$.*

Proof. Since $\|x_\theta - x_{s,\theta}\|_Q^2 = 0\ \forall\theta \in \Omega$, this means that $x_\theta = x_{s,\theta} = (y_{s,\theta}(N), 0)\ \forall\theta \in \Omega$, which implies that $y_{s,\theta} = y_{s,\theta}(N)$ and that $x_{d,\theta} = 0\ \forall\theta \in \Omega$. Then, for the true plant we have that $x_{\theta_t} = (y_{s,\theta_t}, 0)$, which is a steady state with corresponding steady output $y_s = y_{\theta_t} = y_{s,\theta_t}(N)$. Also, given that the prediction of all the models $\theta \in \Omega$ start with $x_\theta = x_{\theta_t}$, they have the same predicted steady output $y_{s,\theta}(N) = y_{s,\theta_t}(N) = y_{\theta_t} = y_s$, which implies that $\Delta\mathbf{u}^* = \{0, \ldots, 0\}$. Now, to prove by contradiction, assume that $y_s \neq y_{sp}^*$. Thus, the optimal control cost corresponding to $(\Delta\mathbf{u}^*, y_{sp}^*)$ is $V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*) = \|y_s - y_{sp}^*\|_S^2$, $\forall\theta \in \Omega$.

Then, since $u^- \in \text{int}(\mathcal{U})$ and we have assumed that $y_s \neq y_{sp}^*$, by virtue of Lemma 7, there exists an admissible control movement Δu such that the predicted steady output of every model $\theta \in \Omega$ can be steered closer to the setpoint by a feasible control sequence $\Delta\tilde{\mathbf{u}} = \{\Delta u, 0, \ldots, 0\}$. Since this control sequence is suboptimal, the associated performance index $V_\theta(x_{\theta_t}, \Delta\tilde{\mathbf{u}}, y_{sp}^*)$ is such that:

$$\begin{aligned}
V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*) &\leq V_\theta(x_{\theta_t}, \Delta\tilde{\mathbf{u}}, y_{sp}^*) \\
&= \sum_{j=0}^{N-1} \left(\|x_\theta(j) - x_{s,\theta}\|_Q^2\right) + \|\Delta u\|_R^2 + \|x_{d,\theta}(N)\|_{P_\theta}^2 + \|y_{s,\theta}(N) - y_{sp}^*\|_S^2 \\
&= \|y_s - y_{s,\theta}^+\|_{Q_s}^2 + \|\Delta u\|_R^2 + \|x_{d,\theta}^+\|_{P_\theta}^2 + \|y_{s,\theta}^+ - y_{sp}^*\|_S^2, \ \forall\theta \in \Omega
\end{aligned}$$

in which $x_{s,\theta} = (y_{s,\theta}(N), 0)$, $y_{s,\theta}(N) = y_{s,\theta}^+ = y_s + B_{s,\theta}\Delta u$ and $x_{d,\theta}^+ = B_{d,\theta}\Delta u$ for all $\theta \in \Omega$. Again, by virtue of Lemma 7, we have that:

$$\begin{aligned}
V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*) &\leq \|y_s - y_{s,\theta}^+\|_{Q_s}^2 + \|\Delta u\|_R^2 + \|x_{d,\theta}^+\|_{P_\theta}^2 + \|y_{s,\theta}^+ - y_{sp}^*\|_S^2 \\
&< \|y_s - y_{sp}^*\|_S^2 \\
&= V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*)
\end{aligned}$$

The strict inequality contradicts the optimality of $V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*)$ and, therefore, the predicted steady output is such that $y_{s,\theta}(N) = y_s = y_{sp}^*$, $\forall\theta \in \Omega$. □

Theorem 3 (Asymptotic stability). *Consider positive definite matrices Q, R, S and P_θ $\forall\theta \in \Omega$, a feasible initial solution $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ and an admissible output setpoint $y_{sp} \in \mathcal{Y}_{sp}$. Then, the application of Algorithm 1 admissibly steers the plantwide state of the true plant to $x_{s,\theta_t} = (y_s, x_d) = (y_{sp}, 0)$, which implies that the output of the true plant is steered to y_{sp}. Moreover, the state x_{s,θ_t} is an asymptotically stable equilibrium point of system* (4.7).

Proof. Convergence: Consider a given time step k. Since there exists a feasible initial condition, by virtue of Lemma 5, problem $P_i(x_k; \Delta\mathbf{u}_k^{[0]}, y_{sp,k}^{[0]})$ is feasible for every agent $i \in \mathbb{I}_{1:M}$ and the sequential solution of this problem is feasible over iterations, thereby resulting in a plantwide solution denoted as $(\Delta\mathbf{u}_k^*, y_{sp,k}^*)$, with the corresponding control cost given as $V_\theta^*(x_k, \Delta\mathbf{u}_k^*, y_{sp,k}^*)$, $\forall\theta \in \Omega$. From (4.29) and (4.30), one obtains a feasible initial solution $(\Delta\mathbf{u}_{k+1}^{[0]}, y_{sp,k+1}^{[0]})$ to be used at the next time step.

Then, by the receding horizon principle, only the first control movement of $\Delta\mathbf{u}_k^*$ is injected into the plant and we move to time step $k+1$, at which a solution of $P_i(x_{k+1}; \Delta\mathbf{u}_{k+1}^{[0]}, y_{sp,k+1}^{[0]})$ is guaranteed to exist by virtue of Corollary 2. Observe that a feasible plantwide suboptimal solution at $k+1$ is $(\Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1}) = (\Delta\mathbf{u}_{k+1}^{[0]}, y_{sp,k+1}^{[0]})$, corresponding to the control cost $\tilde{V}_\theta(x_{\theta,k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$, $\forall\theta \in \Omega$. Note from (4.21) that the predictions of all models $\theta \in \Omega$ start from the state of the true plant θ_t and, consequently, the suboptimal solution $(\Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$ leads to the following sequence of predicted states of the true plant model:

$$\begin{aligned}\{x_{\theta_t}(0|k+1), x_{\theta_t}(1|k+1), \ldots, x_{\theta_t}(N-1|k+1), x_{\theta_t}(N|k+1)\} = \\ \{x_{\theta_t}(1|k), x_{\theta_t}(2|k), \ldots, x_{\theta_t}(N|k), A_{\theta_t}x_{\theta_t}(N|k)\}\end{aligned}$$

In terms of static and dynamic parts, the above sequence of states corresponds to the following sequences of y_{s,θ_t} and x_{d,θ_t}:

$$\begin{aligned}\{y_{s,\theta_t}(0|k+1), y_{s,\theta_t}(1|k+1), \ldots, y_{s,\theta_t}(N-1|k+1), y_{s,\theta_t}(N|k+1)\} = \\ \{y_{s,\theta_t}(1|k), y_{s,\theta_t}(2|k), \ldots, y_{s,\theta_t}(N|k), y_{s,\theta_t}(N|k)\} \\ \{x_{d,\theta_t}(0|k+1), x_{d,\theta_t}(1|k+1), \ldots, x_{d,\theta_t}(N-1|k+1), x_{d,\theta_t}(N|k+1)\} = \\ \{x_{d,\theta_t}(1|k), x_{d,\theta_t}(2|k), \ldots, x_{d,\theta_t}(N|k), F_{\theta_t}x_{d,\theta_t}(N|k)\}\end{aligned}$$

Then, it is clear that $y_{s,\theta_t}(j|k+1) = y_{s,\theta_t}(j+1|k)$ and $x_{d,\theta_t}(j|k+1) = x_{d,\theta_t}(j+1|k)$ and that $x_{s,\theta_t,k+1} = (y_{s,\theta_t}(N|k+1), 0) = (y_{s,\theta_t}(N|k), 0) = x_{s,\theta_t,k}$. Thus, the control cost $\tilde{V}_{\theta_t}(x_{\theta_t,k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$ can be written as follows:

$$\begin{aligned}\tilde{V}_{\theta_t}(x_{\theta_t,k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1}) &= \|x_{\theta_t}(1|k) - x_{s,\theta_t,k}\|_Q^2 + \cdots + \|x_{\theta_t}(N|k) - x_{s,\theta_t,k}\|_Q^2 \\ &+ \|\Delta u^*(1|k)\|_R^2 + \cdots + \|\Delta u^*(N-1|k)\|_R^2 \\ &+ \|F_{\theta_t}x_{d,\theta_t}(N|k)\|_{P_{\theta_t}}^2 + \|y_{s,\theta_t}(N|k) - y_{sp,k}^*\|_S^2 \qquad (4.31)\end{aligned}$$

Observe that:

$$\begin{aligned}\|x_{\theta_t}(N|k) - x_{s,\theta_t,k}\|_Q^2 &= \|y_{s,\theta_t}(N|k) - y_{s,\theta_t}(N|k)\|_{Q_s}^2 + \|x_{d,\theta_t}(N|k)\|_{Q_d}^2 \\ &+ 2\Big(y_{s,\theta_t}(N|k) - y_{s,\theta_t}(N|k)\Big)^T Q_{sd}x_{d,\theta_t}(N|k) \\ &= \|x_{d,\theta_t}(N|k)\|_{Q_d}^2 \qquad (4.32)\end{aligned}$$

Now, combining (4.31) and (4.32) and noticing that, from (4.19), $Q_d + F_{\theta_t}^T P_{\theta_t} F_{\theta_t} = P_{\theta_t}$, it follows that:

$$\begin{aligned}\tilde{V}_{\theta_t}(x_{\theta_t,k+1},\Delta\tilde{\mathbf{u}}_{k+1},\tilde{y}_{sp,k+1}) &= \|x_{\theta_t}(1|k)-x_{s,\theta_t,k}\|_Q^2+\cdots+\|x_{\theta_t}(N-1|k)-x_{s,\theta_t,k}\|_Q^2\\ &\quad+\|x_{d,\theta_t}(N|k)\|_{Q_d}^2+\|\Delta u^*(1|k)\|_R^2+\cdots+\|\Delta u^*(N-1|k)\|_R^2\\ &\quad+\|F_{\theta_t}x_{d,\theta_t}(N|k)\|_{P_{\theta_t}}^2+\|y_{s,\theta_t}(N|k)-y_{sp,k}^*\|_S^2\\ &= \|x_{\theta_t}(1|k)-x_{s,\theta_t,k}\|_Q^2+\cdots+\|x_{\theta_t}(N-1|k)-x_{s,\theta_t,k}\|_Q^2\\ &\quad+\|\Delta u^*(1|k)\|_R^2+\cdots+\|\Delta u^*(N-1|k)\|_R^2\\ &\quad+\|x_{d,\theta_t}(N|k)\|_{P_{\theta_t}}^2+\|y_{s,\theta_t}(N|k)-y_{sp,k}^*\|_S^2\end{aligned}$$

Thus, the following relationship holds:

$$\begin{aligned}\tilde{V}_{\theta_t}(x_{\theta_t,k+1},\Delta\tilde{\mathbf{u}}_{k+1},\tilde{y}_{sp,k+1})-V_{\theta_t}^*(x_{\theta_t,k},\Delta\mathbf{u}_k^*,y_{sp,k}^*) &= -\|x_{\theta_t}(0|k)-x_{s,\theta_t,k}\|_Q^2\\ &\quad-\|\Delta u(0|k)\|_R^2 \qquad (4.33)\end{aligned}$$

Then, it is clear that $\tilde{V}_{\theta_t}(x_{\theta_t,k+1},\Delta\tilde{\mathbf{u}}_{k+1},\tilde{y}_{sp,k+1})\le V_{\theta_t}^*(x_{\theta_t,k},\Delta\mathbf{u}_k^*,y_{sp,k}^*)$ because the right-hand side of (4.33) is non-positive. Note that the robustness constraint (4.27) guarantees that $V_{\theta_t}^*(x_{\theta_t,k+1},\Delta\mathbf{u}_{k+1}^*,y_{sp,k+1}^*)\le\tilde{V}_{\theta_t}(x_{\theta_t,k+1},\Delta\tilde{\mathbf{u}}_{k+1},\tilde{y}_{sp,k+1})$ and we have that:

$$\begin{aligned}V_{\theta_t}^*(x_{\theta_t,k+1},\Delta\mathbf{u}_{k+1}^*,y_{sp,k+1}^*)-V_{\theta_t}^*(x_{\theta_t,k},\Delta\mathbf{u}_k^*,y_{sp,k}^*) &\le -\|x_{\theta_t}(0|k)-x_{s,\theta_t,k}\|_Q^2\\ &\quad-\|\Delta u(0|k)\|_R^2 \qquad (4.34)\end{aligned}$$

From (4.34), we conclude that the sequence that comprises optimal value functions of subsequent time instants is non-increasing, which means that:

$$V_{\theta_t}^*(x_{\theta_t,k+1},\Delta\mathbf{u}_{k+1}^*,y_{sp,k+1}^*)\le V_{\theta_t}^*(x_{\theta_t,k},\Delta\mathbf{u}_k^*,y_{sp,k}^*),\quad\forall k \qquad (4.35)$$

Now, since the control cost is non-increasing and bounded below by zero, it converges, which implies that both sides of (4.34) tend to zero as $k\to\infty$. Consequently, by the positiveness of matrices Q and R, we have that $\lim_{k\to\infty}\|x_{\theta_t}(k)-x_{s,\theta_t,k}\|=0$ and $\lim_{k\to\infty}\|\Delta u(k)\|=0$. Because $\lim_{k\to\infty}\|x_{\theta_t}(k)-x_{s,\theta_t,k}\|=0$, it follows that $\lim_{k\to\infty}\|y_{s,\theta_t}(k)-y_{s,\theta_t}(N|k)\|=0$ and $\lim_{k\to\infty}\|x_{d,\theta_t}(k)\|=0$, implying that $\lim_{k\to\infty}y_{\theta_t}(k)=y_{s,\theta_t}(N|k)$.

Now, in virtue of Lemma 8, we can finally conclude that the plantwide output converges to $y_{\theta_t}(k)=y_{s,\theta_t}(N|k)=y_{sp,k}^*$ as $k\to\infty$.

Stability: Consider an initial state of the true plant $x_{\theta_t,0}=x_{\theta_t}(0)$ and a suboptimal control cost $V_{\theta_t}(x_{\theta_t,0},\Delta\tilde{\mathbf{u}}_0,y_{sp,0})$ computed for $\Delta\tilde{\mathbf{u}}_0=\{0,\ldots,0\}$ and a given $y_{sp,0}\in\mathcal{Y}_{sp}$, then we have that:

$$\begin{aligned}V_{\theta_t}^*(x_{\theta_t,0},\Delta\mathbf{u}_0^*,y_{sp,0}^*)\le V_{\theta_t}(x_{\theta_t,0},\Delta\tilde{\mathbf{u}}_0,y_{sp,0}) &= \|x_{d,\theta_t}(0|0)\|_{P_{\theta_t}}^2+\|y_{s,\theta_t}(0|0)-y_{sp,0}\|_S^2\\ &= \|x_{\theta_t,0}-x_{s,\theta_t,0}\|_Z^2 \qquad (4.36)\end{aligned}$$

in which $x_{s,\theta_t,0} = (y_{s,\theta_t}(0|0), 0)$ and

$$Z = \zeta I_{n_x}, \quad \zeta = \frac{\|x_{d,\theta_t}(0|0)\|^2_{P_{\theta_t}} + \|y_{s,\theta_t}(0|0) - y_{sp,0}\|^2_S}{\|x_{\theta_t,0} - x_{s,\theta_t,0}\|^2}$$

Note that, from the definition of the control cost at a given time step k, the following relationship holds:

$$\|x_{\theta_t,k} - x_{s,\theta_t,k}\|^2_Q \leq V^*_{\theta_t}(x_{\theta_t,k}, \Delta\mathbf{u}^*_k, y^*_{sp,k}) \tag{4.37}$$

Now, from the combination of (4.35), (4.36) and (4.37), it follows that:

$$\|x_{\theta_t,k} - x_{s,\theta_t,k}\|^2_Q \leq \|x_{\theta_t,0} - x_{s,\theta_t,0}\|^2_Z, \quad \forall k > 0 \tag{4.38}$$

Since Q and Z are positive definite matrices, we have that:

$$\lambda_{\min}(Q)\|x_{\theta_t,k} - x_{s,\theta_t,k}\|^2 \leq \zeta\|x_{\theta_t,0} - x_{s,\theta_t,0}\|^2, \quad \forall k > 0 \tag{4.39}$$

Then, for a given $\rho > 0$ such that $\|x_{\theta_t,0} - x_{s,\theta_t,0}\| \leq \rho$, the following relationship holds:

$$\|x_{\theta_t,k} - x_{s,\theta_t,k}\| \leq \epsilon, \quad \forall k > 0 \tag{4.40}$$

in which $\epsilon = \rho\sqrt{\zeta/\lambda_{\min}(Q)}$. Therefore, the closed-loop system is stable.

From the combination of stability and convergences $x_{\theta_t,k} \to x_{s,\theta_t,k}$, $x_{d,\theta_t,k} \to 0$ and $y_{s,\theta_t}(N|k) \to y_{sp,k}$ as $k \to \infty$, it follows that $x_{s,\theta_t} = (y_{sp}, 0)$ is an asymptotically stable equilibrium point of system (4.7). □

4.4 Case study 1: high-purity distillation column

To test the proposed algorithm, we here consider the following model of a high-purity distillation column (RALHAN; BADGWELL, 2000) that has been modified to obtain multiple models with different values of process gains and time constants:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{0.7868(1+\gamma_1)}{(22.98+\beta_1)\,s+1} & \dfrac{-0.6147(1+\gamma_2)}{(22.98+\beta_1)\,s+1} \\[2ex] \dfrac{0.8098(1+\gamma_1)}{(22.98+\beta_2)\,s+1} & \dfrac{-0.982(1+\gamma_2)}{(22.98+\beta_2)\,s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \tag{4.41}$$

The set of models Ω is built considering different values of uncertainty parameters γ_1, γ_2, β_1 and β_2 as given in Table 3.

Table 3 – Parameters of model uncertainty

Model	γ_1	γ_2	β_1	β_2
1	0	0	0	0
2	-0.4	-0.4	10	0
3	-0.4	0.4	10	-10
4	0.4	-0.4	-10	10
5	0.4	0.4	0	-10

All the models have been converted to state-space models, which were discretized using the zero-order hold method with sampling time $T_s = 2$. The system is constrained to $\|u\|_\infty \leq 1.5$ and $\|\Delta u\|_\infty \leq 0.3$. The controller parameters are $N = 2$, $Q_y = 10I_{n_y}$, $Q = C^T Q_y C + 10^{-4} \times I_{n_x}$, $R = 0.2 \times I_{n_u}$ and $S = 10Q_y$. The simulation starts with $y = (0, 0)$, $u = (0, 0)$ and $y_{sp,min} = y_{sp,max} = y_{sp} = (0, 0)$ and, at time step $k = 5$, the output setpoint changes to $y_{sp} = (0.1, 0.13)$. The subsystems are (y_1, u_1) and (y_2, u_2), with weights $w_1 = w_2 = 0.5$. Model 2 is chosen to be the nominal model, while model 5 is the one that represents the true plant.

Remark 8. Note that, since $y_{sp,min} = y_{sp,max} = y_{sp}$, only the setpoint tracking problem is simulated here. This is performed to compare different system responses with respect to the same operating point. The zone control problem will be addressed in the case study presented in Section 4.7.

The results of the proposed robust DMPC (RDMPC) algorithm are compared with the ones obtained by the application of a single-model DMPC that considers only the nominal model. As usual in the DMPC literature, we also present the results of a centralized scheme, corresponding to the robust MPC (RMPC) based on the following optimization problem:

$$(\Delta\mathbf{u}^*, y_{sp}^*) = \arg \min_{\Delta\mathbf{u}, y_{sp}} \quad V_{\theta_n}(x_{\theta_n}, \Delta\mathbf{u}, y_{sp}) \tag{4.42}$$

subject to:

$$x_\theta(0) = x_{\theta_n}, \qquad \theta \in \Omega \tag{4.43}$$

$$x_\theta(j+1) = A_\theta x_\theta(j) + B_\theta \Delta u(j), \quad j \in \mathbb{I}_{0:N-1},\ \theta \in \Omega \tag{4.44}$$

$$u(j) \in \mathcal{U}, \qquad j \in \mathbb{I}_{0:N-1} \tag{4.45}$$

$$\Delta u(j) \in \mathcal{U}_\Delta, \qquad j \in \mathbb{I}_{0:N-1} \tag{4.46}$$

$$y_{sp} \in \mathcal{Y}_{sp} \tag{4.47}$$

$$V_\theta(x_\theta, \Delta\mathbf{u}, y_{sp}) \leq V_\theta(x_\theta, \Delta\tilde{\mathbf{u}}, y_{sp}), \quad \theta \in \Omega \tag{4.48}$$

in which $\Delta\tilde{\mathbf{u}}$ is computed in the same way as $\Delta\mathbf{u}^{[0]}$ in (4.29).

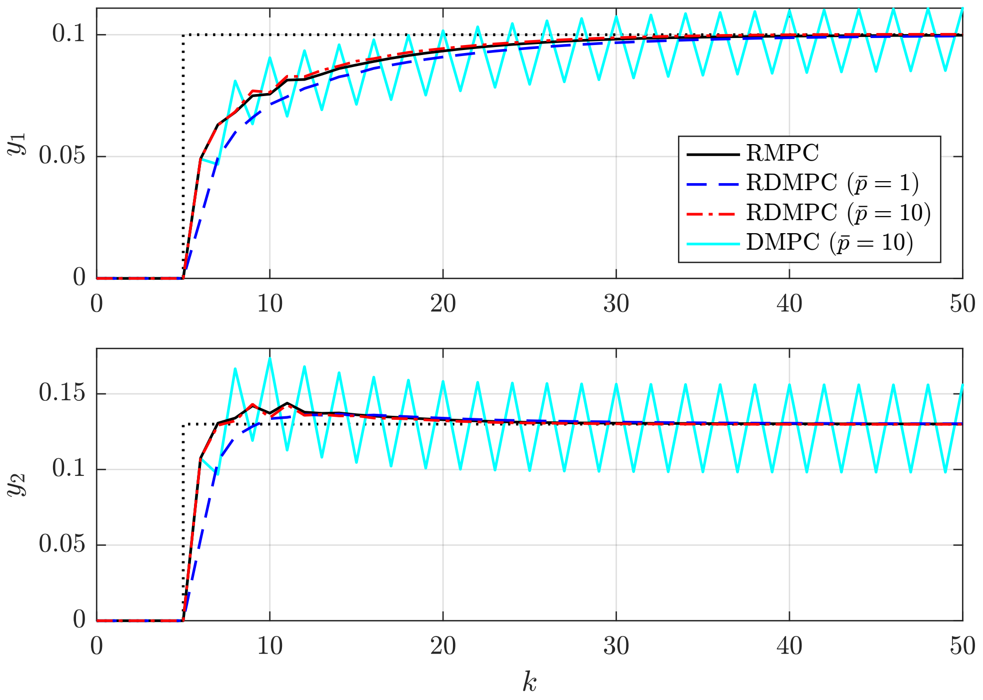

Figure 17 – Outputs of the high-purity distillation column controlled by RMPC, RDMPC and DMPC. Dotted black lines are output setpoints and the remaining lines are described in the legend

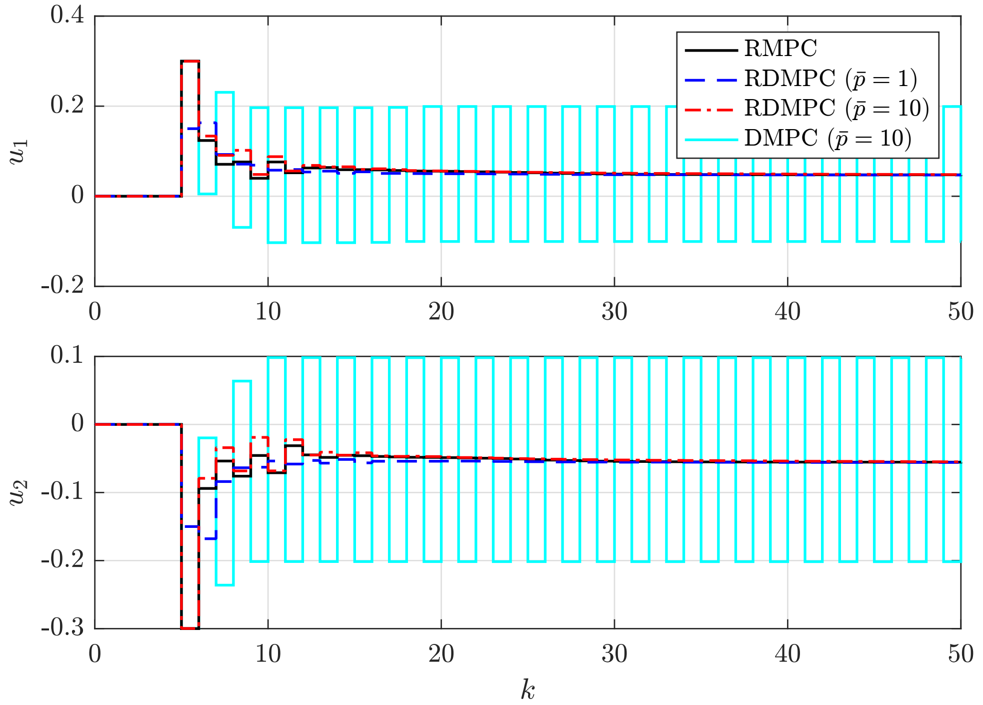

Figure 18 – Inputs of the high-purity distillation column controlled by RMPC, RDMPC and DMPC. The lines are described in the legend

The proposed RDMPC has been applied considering $\bar{p} = 1$ and $\bar{p} = 10$ iterations per time step, while $\bar{p} = 10$ is used in the nominal DMPC. In Figure 17, black dotted lines are the output setpoints and the remaining lines are the system output responses obtained from the application of different controllers, as described in the legend. System inputs are depicted in Figure 18. While the system is steered to the setpoint by the proposed RDMPC for both $\bar{p} = 1$ and $\bar{p} = 10$, it becomes only marginally stable under the application of the nominal DMPC. In fact, the closed-loop system would be unstable under the DMPC if there were no constraints on the control movements, which are clearly active after $k = 10$ (see Figure 18). In comparison with the RDMPC using $\bar{p} = 1$, the RDMPC with $\bar{p} = 10$ produced system responses that are closer to the ones obtained by the centralized RMPC, as expected.

Figure 19 shows the non-increasing behavior of the true plant performance criterion. Note that, unlike nominal DMPC approaches whose performance index should be no better than the one of the centralized MPC, the control cost of the true plant produced by the RDMPC with $\bar{p} = 10$ is sometimes lower. This may occur because, over iterations, the proposed RDMPC improves the performance index computed with the nominal model, which can either improve or worsen the true plant performance. In other words, although V_{θ_n} is always non-increasing over iterations, V_{θ_t} may have different trends. As observed in Figure 20, while both nominal and true plant performance criteria are in general improved at $k = 5$, the true plant performance index increases over iterations at $k = 7$.

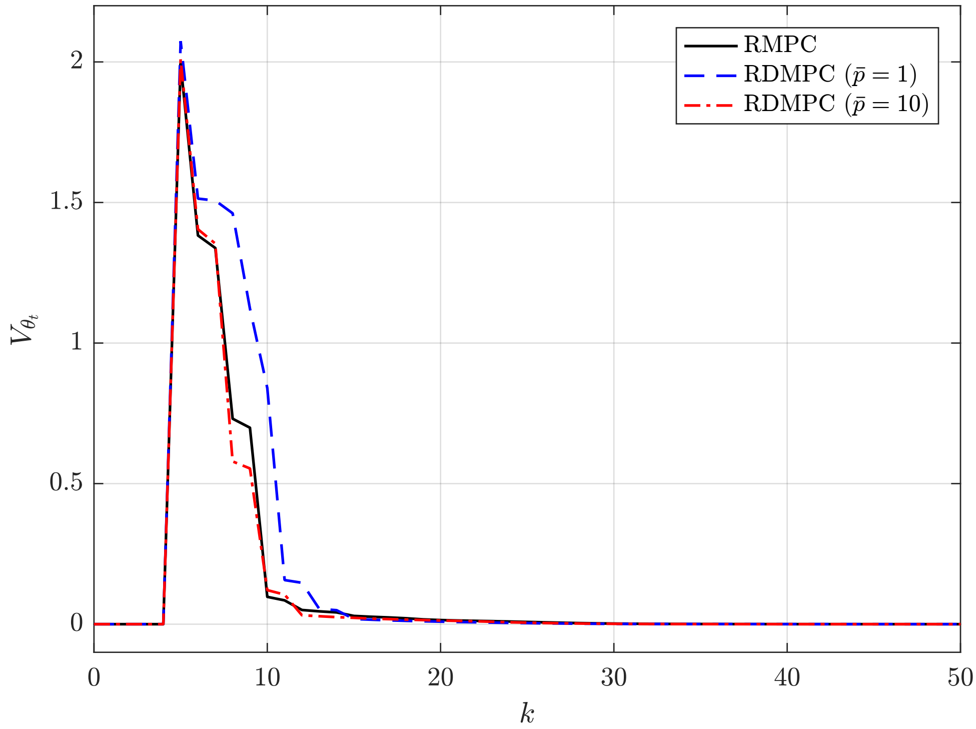

Figure 19 – Performance indexes of the true high-purity distillation column controlled by RMPC and RDMPC. The lines are described in the legend

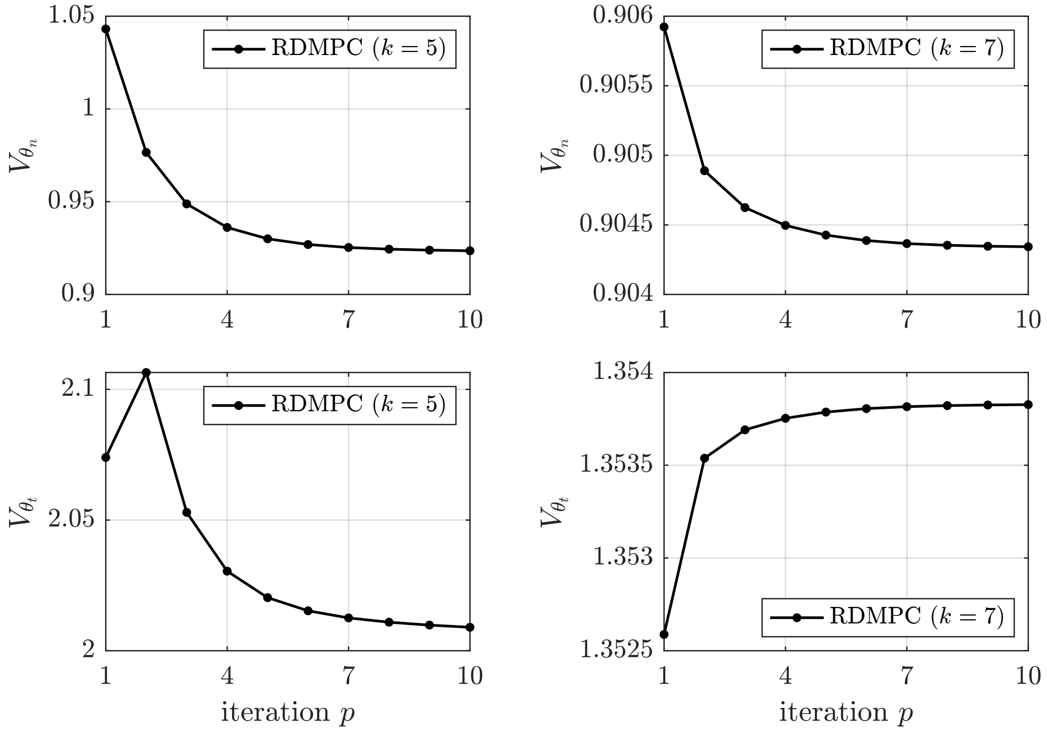

Figure 20 – Nominal and true plant performance criteria produced by RDMPC ($\bar{p} = 10$)

4.5 Improving convergence over iterations

Due to distinct dynamics and magnitude of process gains or even weights in the performance criterion, each subsystem can improve the plantwide performance index to a different extent. This suggests that one can choose the contribution of each subsystem to the global solution such that the resulting performance index of the nominal model is optimized. For this purpose, one can perform the convex combination of $(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*),\ i \in \mathbb{I}_{1:M}$ through an optimization problem in which $w = (w_1, \ldots, w_M)$ is the minimizer of the control cost $V_{\theta_n}(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$. This can be posed as the QP problem $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$ defined as follows:

$$w^* = \arg\min_{w} \quad V_{\theta_n}(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}) \tag{4.49}$$

subject to:

$$x_{\theta_n}(0) = x_{\theta_t} \tag{4.50}$$

$$x_{\theta_n}(j+1) = A_{\theta_n} x_{\theta_n}(j) + \sum_{i=1}^{M} B_{\theta_n,i} \Delta u_i(j), \quad j \in \mathbb{I}_{0:N-1} \tag{4.51}$$

$$(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}) = \sum_{i=1}^{M} w_i (\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*) \tag{4.52}$$

$$\sum_{i=1}^{M} w_i = 1 \tag{4.53}$$

$$w_i \geq w_{i,min}, \quad \forall i \in \mathbb{I}_{1:M} \tag{4.54}$$

in which $w_{i,min} \geq 0,\ \forall i \in \mathbb{I}_{1:M}$ with $\sum_{i=1}^{M} w_{i,min} < 1$. Note that $w_{i,min}$ can be chosen to ensure that the solution of subsystem i always has a contribution to the global solution.

Problem $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$ has M decision variables and, substituting (4.50), (4.51), (4.52) into the objective function, the number of constraints is reduced to one equality and M inequalities. Therefore, since M is typically not very large, the proposed additional QP problem represents a small increase in the required computational cost in comparison to Algorithm 1. Moreover, as in explicit MPC approaches (BEMPORAD *et al.*, 2002), multiparametric quadratic programming tools (OBERDIECK *et al.*, 2016) may be employed so that the computation effort needed to solve the additional QP problem is transferred offline.

We can thus modify Algorithm 1 as follows:

Algorithm 2: Robust cooperative DMPC with improved convergence

```
   Input: $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ and $\mathcal{Y}_{sp}$
1  Set $x = x_{\theta_t}(k)$ and $p = 1$;
2  while stop criteria are not met do
3      for $i \in \mathbb{I}_{1:M}$ do
4          Obtain $(\Delta\mathbf{u}_i^*, y_{sp,i}^*)$ by solving $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$;
5      end
6      Compute $w$ by solving $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$;
7      Compute $(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ through (4.52);
8      $p \leftarrow p + 1$;
9  end
10 Set $(\Delta\mathbf{u}_k, y_{sp,k}) = (\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$;
11 Compute $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ through (4.29) and (4.30);
12 Implement $\Delta u(0|k)$;
13 $k \leftarrow k + 1$;
14 Return to line 1.
```

Since the difference between Algorithm 1 and 2 relies on the weights used in the convex combination defined in (4.28), all the properties proved for Algorithm 1 are retained. Also, because Lemma 6 holds for any convex combination of subsystem solutions and Algorithm 2 has a second optimization step that allows $V_{\theta_n}(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ to decrease, it is easy to see that it improves the convergence of the algorithm.

Remark 9. After the agents communicate their solutions, the computation of w can be performed in two ways: (i) a single coordinating agent computes w and then communicates it to the others, requiring one more communication step or (ii) every agent computes w locally (obviously yielding the same result), which avoids further communication, but requires solving the same problem M times. The choice of strategy may depend on many factors such as the available time to perform iterations, the computing power on hand, the technology employed in the communication between subsystems, etc. For instance, case (ii) may be preferred for time-constrained applications. The reason is that, while the elapsed time for solving the additional QP problem is the same in both cases (i) and (ii) because w is computed in parallel by each agent in (ii), case (i) spends more time during communications. On the other hand, if the time for subsystems to communicate is negligible, case (i) represents the simplest strategy since the additional QP problem is implemented only in a single subsystem.

4.6 Robustifying a nominal cooperative distributed MPC

Note that the problem $P_i(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ given in (4.20)-(4.27) consists of a QCQP problem since constraint (4.27) is quadratic in the decision variables. In order to reduce subsystems' computational cost required to solve their optimal control problems, we propose enforcing the robustness constraint (4.27) not in the MPC problem of each agent, but rather in the last combination of their solutions, i.e. after iterations terminate. This approach allows each agent to account only for the nominal model, resulting in a control scheme that is equivalent to having a nominal cooperative distributed MPC whose solution is robustified before it is applied to the system. Then, provided a feasible initial condition $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ is available, each agent $i \in \mathbb{I}_{1:M}$ solves the QP problem $P_i^{QP}(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ defined as follows:

$$(\Delta\mathbf{u}_i^*, y_{sp,i}^*) = \arg \min_{\Delta\mathbf{u}_i, y_{sp,i}} \quad V_{\theta_n}(x_{\theta_n}, \Delta\mathbf{u}, y_{sp}) \tag{4.55}$$

subject to:

$$x_{\theta_n}(0) = x_{\theta_t} \tag{4.56}$$

$$x_{\theta_n}(j+1) = A_{\theta_n} x_{\theta_n}(j) + \sum_{i=1}^{M} B_{\theta_n,i} \Delta u_i(j), \quad j \in \mathbb{I}_{0:N-1} \tag{4.57}$$

$$\Delta\mathbf{u}_l = \Delta\mathbf{u}_l^{[p-1]}, \qquad l \in \mathbb{I}_{1:M} \setminus i \tag{4.58}$$

$$u(j) \in \mathcal{U}, \qquad j \in \mathbb{I}_{0:N-1} \tag{4.59}$$

$$\Delta u(j) \in \mathcal{U}_\Delta, \qquad j \in \mathbb{I}_{0:N-1} \tag{4.60}$$

$$y_{sp} \in \mathcal{Y}_{sp} \tag{4.61}$$

In the same manner as in Algorithm 2, convergence can be improved by solving problem $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$ defined in (4.49)-(4.52) to perform the convex combination of agents' solutions, producing $(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$. When the maximum number of iterations is achieved or some other stop criterion is met, the agents use their last shared solutions

$(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*)$, $i \in \mathbb{I}_{1:M}$ to recompute $(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ through the following convex combination:

$$(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}) = w_0(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}) + \sum_{i=1}^{M} w_i(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*) \tag{4.62}$$

in which the vector $\tilde{w} = (w_0, w_1, \ldots, w_M)$ is computed as a solution of the following problem $P_{\tilde{w}}(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$:

$$\tilde{w}^* = \arg\min_{\tilde{w}} \quad V_{\theta_n}(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}) \tag{4.63}$$

subject to (4.62) and to:

$$x_\theta(0) = x_{\theta_t}, \qquad \forall\theta \in \Omega \tag{4.64}$$

$$x_\theta(j+1) = A_\theta x_\theta(j) + \sum_{i=1}^{M} B_{\theta,i}\Delta u_i(j), \quad j \in \mathbb{I}_{0:N-1}, \quad \forall\theta \in \Omega \tag{4.65}$$

$$\sum_{i=0}^{M} w_i = 1 \tag{4.66}$$

$$w_i \geq 0, \qquad \forall i \in \mathbb{I}_{0:M} \tag{4.67}$$

$$V_\theta(x_\theta, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}) \leq V_\theta(x_\theta, \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}), \quad \forall\theta \in \Omega \tag{4.68}$$

Note that the computation of $\tilde{w}$ can also be performed in two ways, as discussed in Remark 9.

Algorithm 3: Robust cooperative DMPC with reduced computational cost

```
   Input: $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ and $\mathcal{Y}_{sp}$
1  Set $x = x_{\theta_t}(k)$ and $p = 1$;
2  while stop criteria are not met do
3      for $i \in \mathbb{I}_{1:M}$ do
4          Obtain $(\Delta\mathbf{u}_i^*, y_{sp,i}^*)$ by solving $P_i^{QP}(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$;
5      end
6      Compute $w$ by solving $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$;
7      Compute $(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ through (4.50);
8      $p \leftarrow p + 1$;
9  end
10 Compute $\tilde{w}$ by solving $P_{\tilde{w}}(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$;
11 Recompute $(\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ through (4.62);
12 Set $(\Delta\mathbf{u}_k, y_{sp,k}) = (\Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$;
13 Compute $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ through (4.29) and (4.30);
14 Implement $\Delta u(0|k)$;
15 $k \leftarrow k + 1$;
16 Return to line 1.
```

In the same way as in Algorithm 1 and 2, after the iterative process is terminated, only the first control movement of $\Delta\mathbf{u}_k$ is injected into the plant and this procedure is repeated at the next time step with an initial solution given by (4.29) and (4.30). This is summarized in Algorithm 3.

Note that, if we remove lines 10 and 11, Algorithm 3 reduces to a nominal cooperative distributed MPC in which optimal weights w are computed to improve algorithm convergence. If line 6 is also removed, one obtains a standard nominal cooperative distributed MPC with fixed weights w. This fact suggests that the proposed algorithm can serve as a plug-and-play robustifier module that may be smoothly deployed in a control structure with a running nominal cooperative distributed MPC. Such a feature not only requires minimally invasive and low-cost commissioning, but is also more likely to be accepted by plant operators.

4.6.1 Analyses of the modified algorithm

In what follows, we provide recursive feasibility and convergence analyses of the proposed robust cooperative distributed MPC with reduced computational cost.

Lemma 9 (Recursive feasibility over iterations). *Assume that the initial condition* $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ *is feasible and consider the application of Algorithm 3. Then, problems* $P_i^{QP}(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ *with* $i \in \mathbb{I}_{1:M}$ *and* $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$ *are feasible at iteration* $p = 1$ *and they will remain feasible at any subsequent iteration* $p + j,\ j \in \mathbb{I}_{\geq 1}$*. Moreover, after the iterative procedure terminates, problem* $P_{\tilde{w}}(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$ *will also be feasible.*

Proof. As in the proof of Lemma 5, the feasible initial condition is also a feasible solution to $P_i^{QP}(x; \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ and, thus, every agent $i \in \mathbb{I}_{1:M}$ will have a solution at $p = 1$. Then, $(\Delta\mathbf{u}^{[1]}, y_{sp}^{[1]})$ is obtained through the convex combination defined in (4.50) with w computed as solution of problem $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$, which is always feasible since $w_{i,min} \geq 0,\ \forall i \in \mathbb{I}_{1:M}$ with $\sum_{i=1}^{M} w_{i,min} < 1$.

Next, the computed $(\Delta\mathbf{u}^{[1]}, y_{sp}^{[1]})$ is the initial condition for iteration $p = 2$ that satisfies constrains (4.59)-(4.61) because $\mathcal{U}$, $\mathcal{U}_\Delta$ and $\mathcal{Y}_{sp}$ are convex sets. Consequently, $(\Delta\mathbf{u}^{[1]}, y_{sp}^{[1]})$ serves as a feasible solution to all agents $i \in \mathbb{I}_{1:M}$. Therefore, by induction, we conclude that $P_i^{QP}(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]}), \forall i \in \mathbb{I}_{1:M}$ and $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$ will be feasible at any subsequent iteration $p + j,\ j \in \mathbb{I}_{\geq 1}$.

After the iteration procedure ends, a robust convex combination is performed by solving problem $P_{\tilde{w}}(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$. Observe that this problem always has a suboptimal solution $\tilde{w} = (1, 0, \ldots, 0)$ that clearly satisfy constraints (4.64)-(4.68), which finishes the proof. □

Corollary 3 (Recursive feasibility over time steps)**.** *Consider the application of Algorithm 3. The feasibility of $P_i^{QP}(x; \Delta\mathbf{u}^{[p-1]}, y_{sp}^{[p-1]})$ at iteration $p = 1$ implies the feasibility at every time step $k + j,\ j \in \mathbb{I}_{\geq 1}$.*

Proof. This is a direct result from Lemma 9 and the construction of a feasible initial condition $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ given in (4.29) and (4.30). □

Lemma 10 (Convergence of the algorithm)**.** *In the application of Algorithm 3, the plantwide performance criterion $V_{\theta_n}(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ of the nominal model $\theta_n \in \Omega$ is non-increasing and converges as $p \to \infty$.*

Proof. Since problem $P_{\tilde{w}}(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$ is not solved during iterations, it does not affect the algorithm convergence. Then, we can follow the same steps of the proof of Lemma 6. At an iteration $p + 1$, each agent $i \in \mathbb{I}_{1:M}$ solves problem $P_i^{QP}(x; \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$, resulting in a value function $V_{\theta_n}\left(x_{\theta_n}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right)$ such that $V_{\theta_n}\left(x_{\theta_n}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right) \leq V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}\right)$. From the convex combination defined in (4.50), it follows that:

$$\sum_{i=1}^{M} w_i V_{\theta_n}\left(x_{\theta_n}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right) \leq \sum_{i=1}^{M} w_i V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}\right) = V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}\right)$$

Now, by the convexity of the performance index, we have that:

$$V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p+1]}, y_{sp}^{[p+1]}\right) \leq \sum_{i=1}^{M} w_i V_{\theta_n}\left(x_{\theta_n}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*\right) \leq V_{\theta_n}\left(x_{\theta_n}, \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]}\right)$$

Since the performance index of the nominal model is non-increasing over iterations and given that it is non-negative by construction, it implies that it is bounded below by zero and, therefore, converges. □

Lemma 11 (Auxiliary results)**.** *Consider an admissible output setpoint $y_{sp} \in \mathcal{Y}_{sp}$, an initial solution $(\Delta\mathbf{u}^{[0]}, y_{sp})$ with $\Delta\mathbf{u}^{[0]} = \{0, \ldots, 0\}$ and a given initial state $x = (y_s, 0)$ with $y_s \neq y_{sp}$, and predecessor input $u^- = u_s \in int(\mathcal{U})$. Then, there exists an admissible control movement Δu such that:*

(i) it steers y_s to $y_{s,\theta}^+$ such that the following relationship holds for a positive definite matrix S:

$$\|y_s - y_{s,\theta}^+\|_{Q_s}^2 + \|\Delta u\|_R^2 + \|x_{d,\theta}^+\|_{P_\theta}^2 + \|y_{s,\theta}^+ - y_{sp}\|_S^2 < \|y_s - y_{sp}\|_S^2,\ \forall\theta \in \Omega \tag{4.69}$$

in which $y_{s,\theta}^+ = y_s + B_{s,\theta}\Delta u$ and $x_{d,\theta}^+ = B_{d,\theta}\Delta u$;

(ii) it can be produced by Algorithm 3, which consists of iteratively solving problems $P_i^{QP}(x; \Delta\mathbf{u}^{[p]}, y_{sp}^{[p]})$ and $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$ followed by the solution of problem $P_{\tilde{w}}(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}, \Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$.

Proof. The proof of *(i)* is the same as in Lemma 7. Then, since there exists an admissible control movement Δu that satisfies *(i)*, all the agents $i \in \mathbb{I}_{1:M}$ will have a nontrivial solution for problem $P_i^{QP}(x; \Delta\mathbf{u}^{[0]}, y_{sp})$ which will be combined using optimal weights computed through $P_w(\Delta\bar{\mathbf{u}}_i^*, y_{sp,i}^*, i \in \mathbb{I}_{1:M})$. These solutions are then improved over iterations until some stop criterion is met. At this point, observe that the last agents' solutions do not necessarily satisfies (4.69) for any model $\theta \in \Omega$ other than the nominal one θ_n.

Now, consider a suboptimal sequence of control movements $\Delta\mathbf{u}_i = \{\Delta u_i, 0, \dots, 0\}$ that forms the pair $(\Delta\bar{\mathbf{u}}_i, y_{sp})$, $i \in \mathbb{I}_{1:M}$ and suppose that $\tilde{w} = (1-\beta, \beta/M, \dots, \beta/M)$ with $\beta \in (0,1)$. Then, from (4.62), we have that:

$$\Delta\mathbf{u}^{[1]} = (1-\beta)\Delta\mathbf{u}^{[0]} + \beta/M \sum_{i=1}^{M} \Delta\bar{\mathbf{u}}_i = \beta/M \sum_{i=1}^{M} \Delta\bar{\mathbf{u}}_i = \{\Delta u, 0, \dots, 0\}$$

Thus, assuming $B_{s,\theta}$, $\forall\theta \in \Omega$, has full rank, there exists a sufficiently small $\beta \in (0,1)$ such that the resulting Δu produces a steady output given as $y_{s,\theta}^+ = y_s + B_{s,\theta}\Delta u$ such that $\alpha_{min}\|y_s - y_{sp}\| < \|y_{s,\theta}^+ - y_{sp}\| < \|y_s - y_{sp}\|$ $\forall\theta \in \Omega$, which satisfies (4.69) and, consequently, constraint (4.68) is also satisfied because $V_\theta(x_\theta, \Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}) = \|y_s - y_{sp}\|_S^2, \forall\theta \in \Omega$. This means that problem $P_{\tilde{w}}(\Delta\mathbf{u}^{[0]}, y_{sp}, \Delta\bar{\mathbf{u}}_i^*, y_{sp}, i \in \mathbb{I}_{1:M})$ can generate a vector $\tilde{w}$ that produces an admissible Δu such that (4.69) holds, which finishes the proof of *(ii)*. □

Lemma 12 (Convergence of y_s to y_{sp})**.** *Consider an admissible output setpoint $y_{sp}^* \in \mathcal{Y}_{sp}$ and a given initial state $x_\theta = x_{\theta_t}$, $\forall\theta \in \Omega$, with predecessor input $u^- = u_s \in int(\mathcal{U})$. If the combined solutions resulting from* (4.62) *is* $(\Delta\mathbf{u}^*, y_{sp}^*)$ *such that $\|x_\theta - x_{s,\theta}\|_Q^2 = 0$, then $\|y_{s,\theta}(N) - y_{sp}^*\|_S^2 = 0$, $\forall\theta \in \Omega$.*

Proof. We here follow the same steps as in the proof of Lemma 8. Since $\|x_\theta - x_{s,\theta}\|_Q^2 = 0$, $\forall\theta \in \Omega$, we have that $x_\theta = x_{s,\theta} = (y_{s,\theta}(N), 0)$ $\forall\theta \in \Omega$, which implies that $y_{s,\theta} = y_{s,\theta}(N)$, $x_{d,\theta} = 0$ $\forall\theta \in \Omega$ and that $\Delta\mathbf{u}^* = \{0, \dots, 0\}$. Now, to prove by contradiction, assume that $y_s \neq y_{sp}^*$. Thus, the performance index associated with the solution $(\Delta\mathbf{u}^*, y_{sp}^*)$ is $V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*) = \|y_s - y_{sp}^*\|_S^2$, $\forall\theta \in \Omega$.

Then, by virtue of Lemma 11, since $y_s \neq y_{sp}^*$ and $u^- \in \text{int}(\mathcal{U})$, there exists an admissible control movement Δu such that the predicted steady output of every model $\theta \in \Omega$ can be steered closer to the setpoint by a feasible control sequence $\Delta\tilde{\mathbf{u}} = \{\Delta u, 0, \dots, 0\}$. As also shown in Lemma 11, such a $\Delta\tilde{\mathbf{u}}$ can be produced by a convex combination with weights $\tilde{w}$ computed through problem $P_{\tilde{w}}(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}, \Delta\bar{\mathbf{u}}_i^*, y_{sp}, i \in \mathbb{I}_{1:M})$, in which $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]}) = (\Delta\mathbf{u}^*, y_{sp}^*)$. Then, it can be shown that:

$$\begin{aligned} V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*) &\leq \|y_s - y_{s,\theta}^+\|_{Q_s}^2 + \|\Delta u\|_R^2 + \|x_{d,\theta}^+\|_{P_\theta}^2 + \|y_{s,\theta}^+ - y_{sp}^*\|_S^2 \\ &< \|y_s - y_{sp}^*\|_S^2 \\ &= V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*) \end{aligned}$$

The strict inequality contradicts the optimality of $V_\theta^*(x_{\theta_t}, \Delta\mathbf{u}^*, y_{sp}^*)$ and, therefore, the predicted steady output is such that $y_{s,\theta}(N) = y_s = y_{sp}^*$, $\forall\theta \in \Omega$. □

Theorem 4 (Asymptotic stability)**.** *Consider positive definite matrices Q, R, S and P_θ $\forall\theta \in \Omega$, a feasible initial solution $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ and an admissible output setpoint $y_{sp} \in \mathcal{Y}_{sp}$. Then, the application of Algorithm 3 admissibly steers the plantwide state of the true plant to $x_{s,\theta_t} = (y_s, x_d) = (y_{sp}, 0)$, which implies that the output of the true plant is steered to y_{sp}. Moreover, the state x_{s,θ_t} is an asymptotically stable equilibrium point of system* (4.7)*.*

Proof. Convergence: We proceed by following the same arguments as in the proof of Theorem 3. In virtue of Lemma 9, the existence of a feasible initial condition $(\Delta\mathbf{u}^{[0]}, y_{sp}^{[0]})$ at time step k implies that a plantwide solution $(\Delta\mathbf{u}_k^*, y_{sp,k}^*)$ can be obtained, which corresponds to $V_\theta^*(x_k, \Delta\mathbf{u}_k^*, y_{sp,k}^*)$, $\forall\theta \in \Omega$. Also, from (4.29) and (4.30), one obtains a feasible initial condition $(\Delta\mathbf{u}_{k+1}^{[0]}, y_{sp,k+1}^{[0]})$ to be used in the next time step.

By following the receding horizon principle, only the first control movement of $\Delta\mathbf{u}_k^*$ is injected into the plant and we move to time step $k+1$, at which problems $P_i^{QP}(x_{k+1}; \Delta\mathbf{u}_{k+1}^{[0]}, y_{sp,k+1}^{[0]})$ and $P_{\tilde{w}}(\Delta\mathbf{u}_{k+1}^{[0]}, y_{sp,k+1}^{[0]}, \Delta\bar{\mathbf{u}}_{i,k+1}^*, y_{sp,i,k+1}^*, i \in \mathbb{I}_{1:M})$ are feasible by virtue of Corollary 3. Considering $\tilde{w} = \{1, 0, \ldots, 0\}$, a plantwide suboptimal solution is $(\Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1}) = (\Delta\mathbf{u}_{k+1}^{[0]}, y_{sp,k+1}^{[0]})$, corresponding to the performance index $\tilde{V}_\theta(x_{\theta,k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$, $\forall\theta \in \Omega$. Note that, from (4.64), the predictions of all models $\theta \in \Omega$ start from the state of the true plant θ_t and, consequently, for the true plant, the suboptimal solution $(\Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$ leads to $y_{s,\theta_t}(j|k+1) = y_{s,\theta_t}(j+1|k)$ and $x_{d,\theta_t}(j|k+1) = x_{d,\theta_t}(j+1|k)$, as already shown in the proof of Theorem 3. Thus, the following relationship holds:

$$\begin{aligned}\tilde{V}_{\theta_t}(x_{\theta_t,k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1}) - V_{\theta_t}^*(x_{\theta_t,k}, \Delta\mathbf{u}_k^*, y_{sp,k}^*) = &- \|x_{\theta_t}(0|k) - x_{s,\theta_t,k}\|_Q^2 \\ &- \|\Delta u(0|k)\|_R^2 \end{aligned} \tag{4.70}$$

Then, since $\tilde{V}_{\theta_t}(x_{\theta_t,k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1}) \leq V_{\theta_t}^*(x_{\theta_t,k}, \Delta\mathbf{u}_k^*, y_{sp,k}^*)$ and noting that $V_{\theta_t}^*(x_{\theta_t,k+1}, \Delta\mathbf{u}_{k+1}^*, y_{sp,k+1}^*) \leq \tilde{V}_{\theta_t}(x_{\theta_t,k+1}, \Delta\tilde{\mathbf{u}}_{k+1}, \tilde{y}_{sp,k+1})$ is ensured by the robustness constraint (4.68), we have that:

$$\begin{aligned}V_{\theta_t}^*(x_{\theta_t,k+1}, \Delta\mathbf{u}_{k+1}^*, y_{sp,k+1}^*) - V_{\theta_t}^*(x_{\theta_t,k}, \Delta\mathbf{u}_k^*, y_{sp,k}^*) \leq &- \|x_{\theta_t}(0|k) - x_{s,\theta_t,k}\|_Q^2 \\ &- \|\Delta u(0|k)\|_R^2 \end{aligned} \tag{4.71}$$

Consequently, the sequence that comprises optimal value functions of subsequent time instants is non-increasing, which means that:

$$V_{\theta_t}^*(x_{\theta_t,k+1}, \Delta\mathbf{u}_{k+1}^*, y_{sp,k+1}^*) \leq V_{\theta_t}^*(x_{\theta_t,k}, \Delta\mathbf{u}_k^*, y_{sp,k}^*), \quad \forall k \tag{4.72}$$

Now, since the control cost is non-increasing and bounded below by zero, it converges, which implies that both sides of (4.71) tend to zero as $k \to \infty$. Consequently, by the positiveness of matrices Q and R, we have that $\lim_{k\to\infty}\|x_{\theta_t}(k) - x_{s,\theta_t,k}\| = 0$ and $\lim_{k\to\infty}\|\Delta u(k)\| = 0$. Since $\lim_{k\to\infty}\|x_{\theta_t}(k) - x_{s,\theta_t,k}\| = 0$, it follows that $\lim_{k\to\infty}\|y_{s,\theta_t}(k) - y_{s,\theta_t}(N|k)\| = 0$ and $\lim_{k\to\infty}\|x_{d,\theta_t}(k)\| = 0$, implying that $\lim_{k\to\infty} y_{\theta_t}(k) = y_{s,\theta_t}(N|k)$.

Now, in virtue of Lemma 12, we can finally conclude that the plantwide output converges to $y_{\theta_t}(k) = y_{s,\theta_t}(N|k) = y^*_{sp,k}$ as $k \to \infty$.

Stability: Consider an initial state of the true plant $x_{\theta_t,0} = x_{\theta_t}(0)$ and a suboptimal control cost $V_{\theta_t}(x_{\theta_t,0}, \Delta\tilde{\mathbf{u}}_0, y_{sp,0})$ computed for $\Delta\tilde{\mathbf{u}}_0 = \{0, \ldots, 0\}$ and a given $y_{sp,0} \in \mathcal{Y}_{sp}$. Then, by following the same steps as in the proof of Theorem 3, we can state that, for a given $\rho > 0$ such that $\|x_{\theta_t,0} - x_{s,\theta_t,0}\| \leq \rho$ with $x_{s,\theta_t,0} = (y_{s,\theta_t}(0|0), 0)$, the following relationship holds:

$$\|x_{\theta_t,k} - x_{s,\theta_t,k}\| \leq \epsilon, \quad \forall k > 0 \tag{4.73}$$

in which $\epsilon = \rho\sqrt{\zeta/\lambda_{\min}(Q)}$ and

$$\zeta = \frac{\|x_{d,\theta_t}(0|0)\|^2_{P_{\theta_t}} + \|y_{s,\theta_t}(0|0) - y_{sp,0}\|^2_S}{\|x_{\theta_t,0} - x_{s,\theta_t,0}\|^2}$$

Therefore, the closed-loop system is stable and, from the combination of stability and convergences $x_{\theta_t,k} \to x_{s,\theta_t,k}$, $x_{d,\theta_t,k} \to 0$ and $y_{s,\theta_t}(N|k) \to y_{sp,k}$ as $k \to \infty$, we conclude that $x_{s,\theta_t} = (y_{sp}, 0)$ is an asymptotically stable equilibrium point of system (4.7). □

4.7 Case study 2: two reactors-separator process

In this case study, we consider the three-unit process represented in the schematic diagram depicted in Figure 21. This plant consists of two continuous stirred-tank reactors and a flash tank (separator) (LIU; de la PEÑA; CHRISTOFIDES, 2009). The CSTRs operate in series and are fed with pure reactant A through streams F_{10} and F_{20}. The reactions $A \to B$ and $B \to C$ take place in the reactors, in which B is the desired product and C is an undesired side-product. The effluent from the second CSTR is fed to the flash tank whose overhead vapor is condensed and partially recycled to the first CSTR. The remaining condensed overhead vapor is purged and the bottom stream is the purified product.

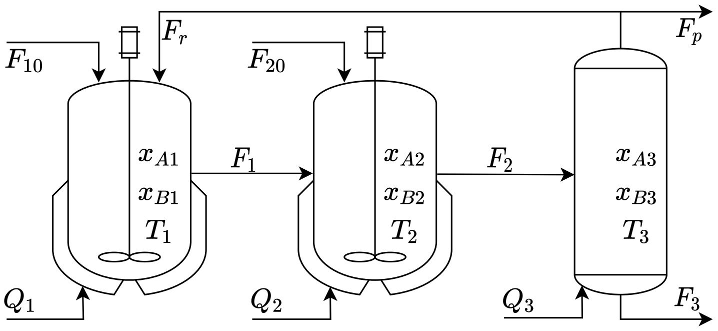

Figure 21 – Schematic diagram of the two reactors-separator process

Under the assumption of static holdup in the three vessels, the overall dynamics of this system are described by 9 differential equations presented in Liu, de la Peña and Christofides (2009). In this work, we use the model parameters given in Li and Swartz (2019, Table 4). The model has been linearized in the operating points corresponding to the steady states and inputs given in Tables 4 and 5, respectively.

Table 4 – Steady-state values for different operating points of the two reactors-separator process

Steady states	Operating point		
	OP1	OP2	OP3
x_{A1}	0.2628	0.2266	0.3028
x_{B1}	0.3968	0.3747	0.4055
T_1	337.04	344.57	335.96
x_{A2}	0.1049	0.0816	0.1364
x_{B2}	0.4043	0.3623	0.4401
T_2	344.45	353.52	342.35
x_{A3}	0.0569	0.0413	0.0740
x_{B3}	0.4755	0.4349	0.5208
T_3	346.53	356.35	344.39

Table 5 – Steady-input values for different operating points of the two reactors-separator process

Steady inputs	Operating point		
	OP1	OP2	OP3
F_{10}	8.3	7	10
Q_1	10	20	14
F_{20}	0.5	0.3	0.7
Q_2	10	16	8
F_r	4	4	5
Q_3	10	12	12

The obtained linear state-space models have been discretized using the zero-order hold method and sampling time $T_s = 2$. In this case study, we use the nonlinear model to simulate the real plant with perfect state measurement and consider the nominal model as the one obtained in the operating point OP1. Upper and lower bounds of inputs are, in absolute values, $u_{max} = (15, 15, 5, 15, 7, 15)$ and $u_{min} = (0, 0, 0, 0, 0, 0)$, respectively, and the input increments are limited to $\Delta u_{max} = 0.1u_{max}$. The parameters used in the controllers are $N = 4$, $Q_y = diag(10^4, 10^4, 1, 10^4, 10^4, 1, 10^4, 10^4, 1)$, $Q = C^T Q_y C + 10^{-4} \times I_{n_x}$, $R = I_{n_u}$ and $S = 10^2 \times I_{n_y}$. Distributed models are obtaining by considering each unit as a subsystem, resulting in $y_1 = (x_{A1}, x_{B1}, T_1)$, $u_1 = (F_{10}, Q_1)$, $y_2 = (x_{A2}, x_{B2}, T_2)$, $u_2 = (F_{20}, Q_2)$, $y_3 = (x_{A3}, x_{B3}, T_3)$, $u_3 = (F_r, Q_3)$. Throughout the subsequent discussions of results, for conciseness, A1, A2 and A3 will refer to Algorithms 1, 2 and 3, respectively. During the simulations, A1, A2 and A3 perform only a single iteration per time step.

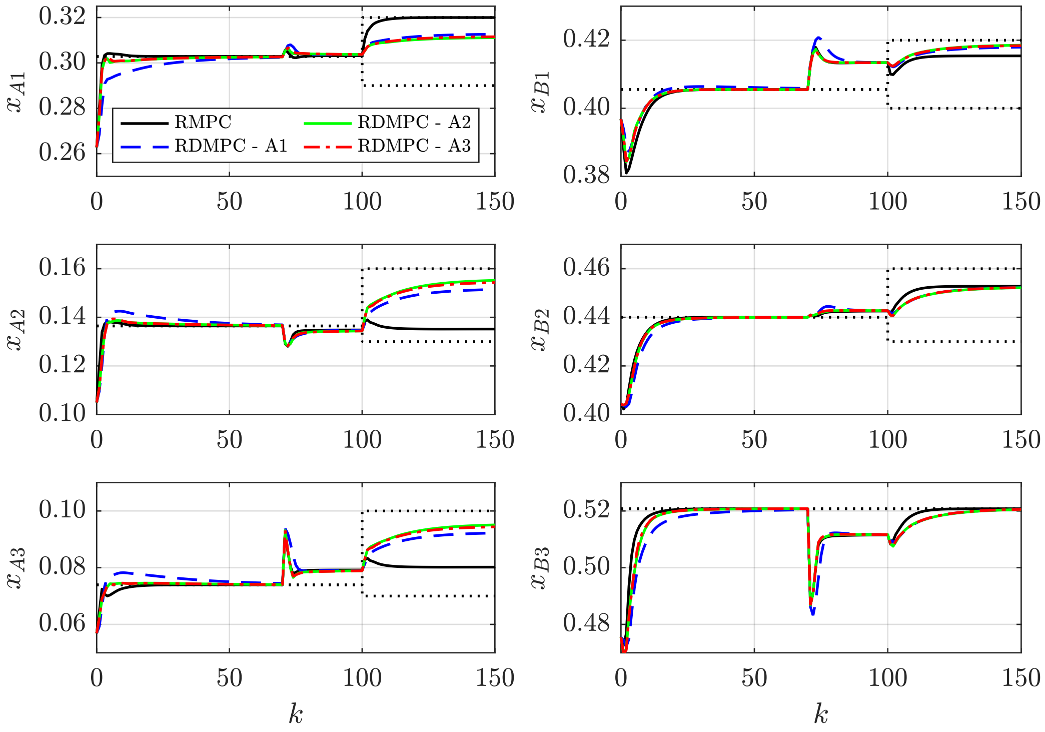

Figure 22 – Concentrations of the two reactors-separator process controlled by RMPC and RDMPC algorithms A1, A2 and A3. Dotted black lines are output setpoints and control zones. The remaining lines are described in the legend

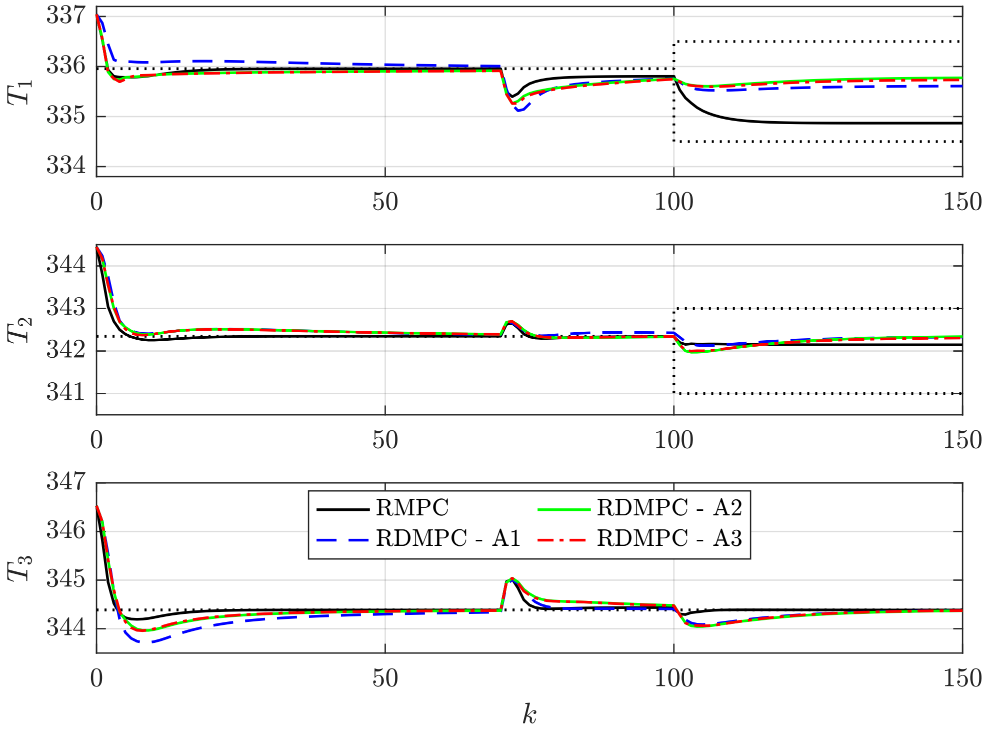

Figure 23 – Temperatures of the two reactors-separator process controlled by RMPC and RDMPC algorithms A1, A2 and A3. Dotted black lines are output setpoints and control zones. The remaining lines are described in the legend

The system is initially at operating point OP1 and the output setpoint corresponds to the steady state of OP3 (see Table 4). Output responses are shown in Figures 22 and 23 in which black dotted lines represents either output setpoints or limits of control zones. Computed manipulated variables are given in Figure 24 in which black dotted-lines corresponds to input constraints. At time instant $k = 70$, an unmeasured step input disturbance of amplitude -3 is introduced in F_r, which makes F_r and Q_1 saturate at their upper limits, leading to an offset in the controlled variables. Then, we switch to the zone control mode with $y_{min} = (0.29, 0.40, 334.5, 0.13, 0.43, 341, 0.07, 0.5208, 344.39)$ and $y_{max} = (0.32, 0.42, 336.5, 0.16, 0.46, 343, 0.10, 0.5208, 344.39)$. Note that x_{B3} is still controlled at a setpoint to simulate a potential desired product specification. By increasing the degrees of freedom, the zone control mode rejects the disturbance and allows important variables to return to their setpoints, while the others are steered into the control zone, in which there is no preference for a specific setpoint. Compared with A1, algorithms A2 and A3 produced system responses that are closer to the ones obtained by the centralized controller only during the setpoint tracking mode. This was not the case under the zone control strategy because A2 and A3 allow for different combinations of agents' solutions, which generates setpoints that are only required to be inside the control zones.

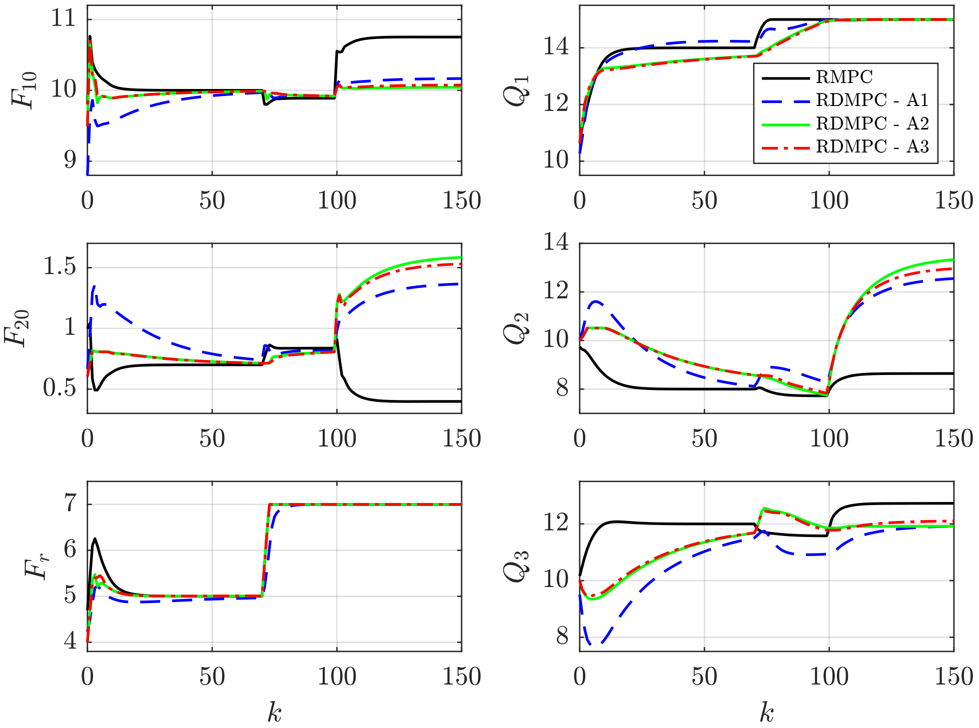

Figure 24 – Inputs of the two reactors-separator process controlled by RMPC and RDMPC algorithms A1, A2 and A3. The lines are described in the legend

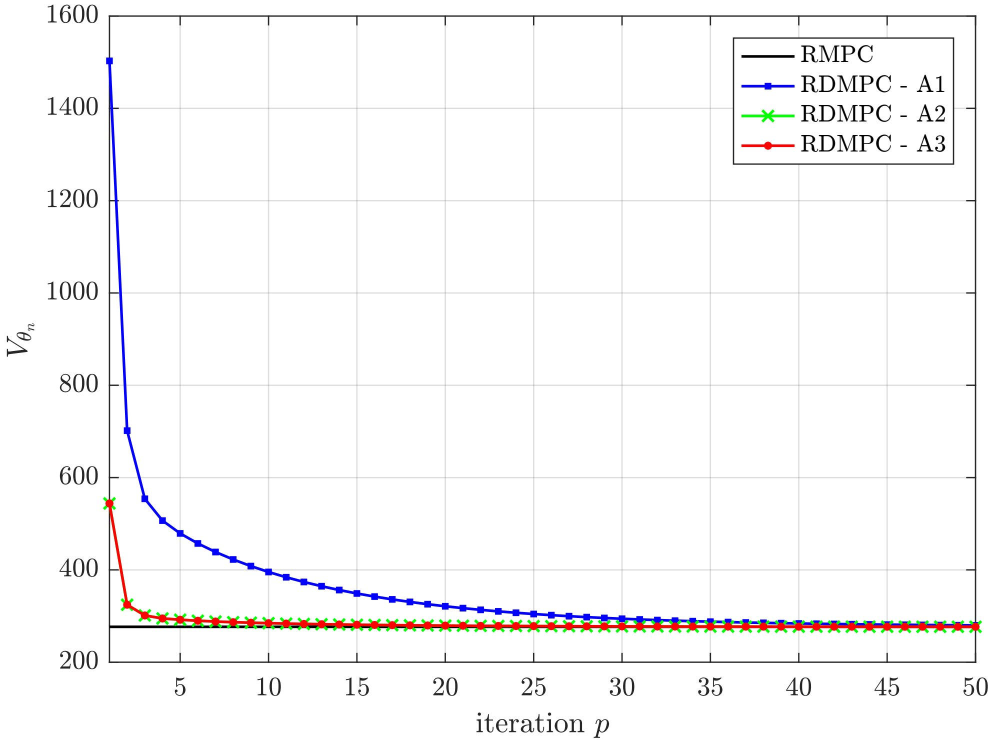

Figure 25 – Convergence of the performance criteria obtained with A1, A2 and A3 when performing 50 iterations at $k = 0$

Unlike A1 which performs the convex combination of solutions with fixed weights at each iteration, optimizing weights are computed in A2 and A3, allowing the performance criterion of the nominal model to decrease. To illustrate the difference of convergence between the proposed algorithms, we show in Figure 25 the performance criteria obtained with A1, A2 and A3 at time step $k = 0$. Here, we simulated 50 iterations instead of a single one. Note that, for a given iteration p before convergence, the performance criteria of the nominal model produced by A2 and A3 are the closest to the one obtained with the centralized controller. Moreover, for all the algorithms, V_{θ_n} is non-increasing over iterations and converges, as stated in Lemma 6.

To compare the computational burden of the proposed algorithms, the CPU time spent by all the agents has been measured at each time step and an average has been taken to represent the CPU time spent per agent. In A2 and A3, we have considered the worst case in which all the agents solve the optimization problem to compute the weights for the convex combination of solutions (see Remark 9). Simulations have been performed using MATLAB® in a computer equipped with a 2.00 GHz Intel® Core™ i7-4510U CPU. All optimal control problems have been built using CasADi (ANDERSSON *et al.*, 2019) and QP and NLP problems have been solved with qpOASES (FERREAU *et al.*, 2014) and Ipopt (WÄCHTER; BIEGLER, 2006), respectively. Since implementations were not intended to minimize computational cost, QCQP problems were solved via NLP. Also, the time that agents would spend communicating their solutions is not considered here. In this sense, the analysis of results will be focused on the type and dimension of the optimization problems that each algorithm solves.

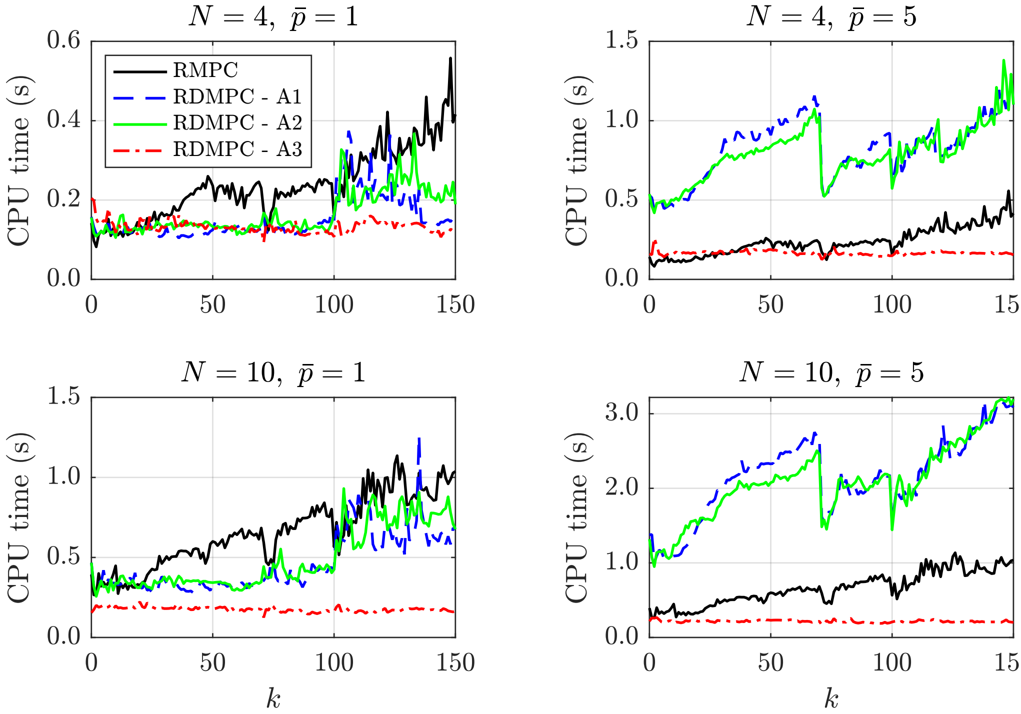

Figure 26 – CPU time spent over the simulation for different N and $\bar{p}$

Table 6 – Total CPU time for different N and $\bar{p}$

Controller	Total CPU time (s)	
	$N = 4$	$N = 10$
RMPC	35.87	99.91
RDMPC-A1 ($\bar{p} = 1$)	23.06	69.62
RDMPC-A2 ($\bar{p} = 1$)	25.12	73.76
RDMPC-A3 ($\bar{p} = 1$)	19.95	26.54
RDMPC-A1 ($\bar{p} = 5$)	124.29	323.92
RDMPC-A2 ($\bar{p} = 5$)	119.03	315.37
RDMPC-A3 ($\bar{p} = 5$)	25.10	32.68

Figure 26 shows CPU times for different prediction horizons and number of iterations, while Table 6 gives the total amount of CPU time spent throughout simulations. As expected, CPU time increases with both the prediction horizon length and the number of iterations. Hence, the differences between the CPU times spent by the proposed algorithms are related to the extent that each one is affected by N and $\bar{p}$. For instance, A1 and A2 have the optimal control problem posed as an NLP problem and, thus, are more sensitive to N in comparison to A3, whose control problem is posed as a QP problem. Compared with the convex NLP problems solved by the agents in A1 and A2, the QP problems solved in A3 account only for the nominal model and have a much lower computational cost. This also explains why the difference between the CPU times spent by A1 and A2 is very small. The additional QP problem solved in A2 has a relatively low computational cost, especially for this small-size example with only 3 subsystems.

As observed in Table 6, while A1 produced a lower total CPU time in comparison to A2 for $\bar{p} = 1$, the opposite holds for $\bar{p} = 5$. This may occur when the optimal convex combination computed in A2 results in a plantwide solution that improves the initial guess provided to the NLP solver at the next iteration. Concerning the effect of the number of iterations, while the amount of NLP problems solved in A1 and A2 increases with $\bar{p}$, A3 solves only one NLP problem per time step, no matter how many iterations are performed. Beside the fact that the NLP problem is solved in A3 to robustify the convex combination of agents' solutions only after iterations terminate, it consists of a much simpler problem with only $M + 1$ scalars as decision variables. This explains why A3 is less sensitive to the control horizon length N in comparison with A1 and A2.

4.8 Conclusion

In this chapter, we presented three algorithms for a robust cooperative distributed MPC based on a multi-model approach. The proposed strategies are built upon a finite family of OPOM models and are suitable for dealing with both setpoint tracking and the zone control problem. The use of velocity-form models avoids the need for estimating disturbances and an upper layer that computes reachable steady states to be used as references by the MPC. From a practical point of view, a multi-model plant description may represent a more tractable and implementable strategy since a set of models obtained at different operating conditions can fairly cover the dynamics of moderately nonlinear systems. Also, they are easier to obtain in comparison to other approaches that are based on polytopic plants or employ robust invariant sets.

In Algorithm 1, the plantwide solution is obtained by means of a conventional convex combination with fixed weights, while optimal ones are computed in Algorithm 2, which improves convergence over iterations. Regarding Algorithm 3, it can be viewed as a usual nominal distributed MPC with improved convergence through optimal convex combinations and with a solution robustification step after the iterative procedure. All these algorithms enjoy properties such as recursive feasibility, convergence, and asymptotic stability.

In case study 1, it was shown an example in which a nominal DMPC could not drive the system to the desired setpoint, which, in contrast, was accomplished by the proposed Algorithm 1. Due to active constraints on input increments, only marginal stability is achieved under the application of the nominal DMPC, which would lead to an unstable closed-loop behavior in the absence of these constraints. Results also showed that optimizing for the nominal model can either decrease or increase the performance index of the true plant over iterations. This means that, in case of a large difference between the nominal model and the true plant, the suboptimal (from the nominal model viewpoint) robust DMPC with a single iteration may perform better than a robust centralized MPC, which does not occur in nominal DMPC formulations. However, since we do not know which model represents the true plant, this conservative and suboptimal closed-loop performance is expected, which is, in fact, the trade-off that most robust MPC approaches exhibit.

Both setpoint tracking and zone control problems were explored in case study 2, showing the extended applicability of the proposed algorithms in comparison to robust DMPC approaches available in the literature that usually do not account for the zone control problem, very common in the process industry. This case study also shows that Algorithms 2 and 3 benefit from the computation of optimal convex combinations and have faster convergences of the nominal performance index over Algorithm 1. Moreover, measurements of CPU time spent by the algorithms along simulations suggest that Algorithm 3 is the least sensitive to the increase of the prediction horizon length and number of iterations. Considering these results and that Algorithm 3 can be viewed as a robustification module applied to a nominal DMPC, it can be said that Algorithm 3 is the most advantageous for a practical application.

Chapter 5

Conclusions and future work directions

5.1 Summary of contributions

In this work, stabilizing MPC approaches based on the output prediction-oriented model are proposed. With the purpose of reducing the complexity involved in the construction of the OPOM model, a straightforward method for building an OPOM-like model is proposed. The resulting state-space model in the velocity form of inputs is named as the new OPOM since the main feature of OPOM is retained, namely the output prediction at steady state embedded in the state vector. Unlike the traditional OPOM whose construction is based on transfer functions, the new OPOM is built from a conventional state-space model, which corresponds to a more usual representation of MIMO systems. Representing a generalization of the traditional OPOM, the approach presented in this work deals with systems containing stable, integrating and unstable poles as well as time delays and pole multiplicity.

The proposed new OPOM has additional features that are potentially interesting for practical applications. For instance, by using the final formulas to compute system matrices, one can easily automate the construction of the new OPOM in a computer program, which is an advantage for building an industrial package. Moreover, it was shown that the state vector of the new OPOM can be easily obtained when the model used to build the new OPOM has measured states. This property allows a straightforward integration between the realigned model and the new OPOM to obtain output feedback. In case only a state estimate of the original model is available, it can be converted to produce an estimate of the state vector of the new OPOM. This avoids the need for a state observer specifically designed for the new OPOM, representing an important advantage over the traditional OPOM, which requires a tailor-designed state observer. In addition to its applicability to existing IHMPC formulations with OPOM, the new OPOM can also serve as a general framework for future developments of IHMPC controllers based on velocity models.

Another contribution of this work concerns the proposal of a dual-mode MPC with OPOM that results from the combination of the approaches presented in Rodrigues and Odloak (2003b) and Limon *et al.* (2008). In the proposed controller, either the new OPOM or the traditional one can be used. Since an MPC with OPOM that employs a terminal control law (apart from the trivial one) was still lacking, the proposed dual-mode strategy fills this gap. A terminal control law stabilizes the dynamic part of OPOM, which is then used in the characterization of steady outputs and inputs. This approach avoids the parametrization of system equilibria proposed by Limon *et al.* (2008), which is usually very sensitive to model errors. Hence, even in the presence of plant-model mismatch and constant unmeasured disturbances, the method proposed in this work produces artificial references that are consistent with the real plant steady state. Such a feature allows establishing a well-posed performance index that, allied to an unbiased prediction at steady state through the use of a velocity model, results in dual-mode MPC with embedded integral action. Theoretical properties such as recursive feasibility and closed-loop convergence were proved by means of standard methods.

From an application viewpoint, the dual-mode MPC with OPOM presented here explicitly deals with constrained input increments and allows for a simpler out-of-the-box implementation since, unlike the MPC proposed by Limon *et al.* (2008), no extra ingredient to produce offset-free control is required. Moreover, if the new OPOM is used and built from the realigned model, it provides an easy way towards obtaining an output-feedback controller. Still considering practical aspects and following the same philosophy adopted in most MPC formulations with OPOM, the proposed dual-mode MPC is also extended for dealing with output control zones and input targets. In this approach, steady-state objectives are translated into an offset cost function. It is proved that the proposed control strategy steers the outputs into their control zones and the inputs to their desired targets if this is an admissible plant equilibrium. Otherwise, the system is steered to the operating point corresponding to the minimum of the offset cost function. Unlike existing works of MPC with OPOM that establishes minimum values for S_y and S_u (corresponding to weighting matrices of offset in the outputs and inputs, respectively) for ensuring system convergence to the desired operating point (assumed admissible), here we prove that this convergence is guaranteed for S_y and S_u being only positive definite weighting matrices. This represents a way more relaxed requirement.

Motivated by recently published cooperative distributed MPC formulations with OPOM (SANTANA; MARTINS; ODLOAK, 2020b; SARAPKA; MARTINS; ODLOAK, 2021) and inspired by the robust IHMPC of Odloak (2004), this work also presents three different algorithms for a robust cooperative distributed MPC. The proposed distributed control strategies are based on the multi-model uncertainty paradigm and are suitable for dealing with both setpoint and zone control problems. Algorithm 1 can be viewed as a distributed version of the robust IHMPC proposed in Odloak (2004) but with some modifications in the performance index regarding the penalization of states and the explicit use of artificial references, which is equivalent to the inclusion of slack variables. Recursive problem feasibility, algorithm convergence and asymptotic stability are proved. As in the dual-mode MPC with OPOM, the convergence of system output to the setpoint (assumed admissible) is ensured by simply using a positive-definite offset weighting matrix.

As an enhanced version of Algorithm 1, the idea presented in Algorithm 2 consists of computing optimizing weights to perform the convex combination of subsystems' solutions. The additional optimization problem produces the best performance index out of the available solutions of agents, which improves algorithm convergence, possibly reducing the number of iterations. To the author's knowledge, this approach had not been proposed in the literature so far and, thus, may represent a promising strategy to speed up the convergence of existing distributed MPCs.

A different approach concerning the robust cooperative distributed MPC is proposed in Algorithm 3. In this control strategy, the robustness constraint is removed from the control problem solved by the agents and transferred to an optimization problem responsible for computing optimizing weights after the last iteration. By following this approach, the control problem solved by each agent is simplified to a nominal distributed MPC, which thus can be posed as a QP problem, reducing the computational burden. Consequently, while Algorithms 1 and 2 solve NLP problems at every single iteration, in the proposed Algorithm 3, an NLP problem is solved only when the iteration procedure terminates. As discussed, optimizing convex combination weights can be computed either locally by each agent or by a single coordinating agent that further communicates the result to the others. Therefore, due to the nature and number of optimizations solved by each algorithm, CPU time spent by Algorithm 3 tends to be less sensitive to problem dimension and number of iterations, as was in fact observed in the simulated results. This represents an important advantage in practical applications with limited computation time, in which solving various NLP problems can be prohibitive.

Finally, an interesting feature of Algorithm 3 should be highlighted. In this proposed strategy, suitable weights used to combine agents' solutions are computed so that the robustness property is enforced only after the very last iteration. In this sense, Algorithm 3 can be viewed as a robustifier module coupled to a nominal distributed MPC. Such a feature paves the way for designing an update package that robustifies a running distributed MPC, which represents a less invasive implementation and possibly favors industrial applications since practitioners are usually reluctant to change functional existing control structures.

5.2 Suggestions for future works

The following ideas can be considered for future research:

- Problem infeasibilities may arise when the dual-mode MPC with OPOM is applied in the presence of plant-model mismatch or disturbances since, in this case, the property of nominal recursive feasibility is lost. Hence, the use of soft terminal constraints represents a useful tool for improving the feasibility of the proposed controller. However, instead of penalizing slack variables using only quadratic norms, this can be performed by applying the concept of exact penalty functions (ZEILINGER; MORARI; JONES, 2014). Such a strategy guarantees the minimum use of slack variables, recovering stability properties whenever the hard terminal constraint can be enforced.

- Although optimizing input targets are handled by the proposed dual-mode MPC with OPOM, this represents only a steady-state economic objective. Thus, the dual-mode MPC with OPOM proposed in this work can be further extended for coping with economic objectives during transients as well. This can be performed by considering the one-layer MPC + RTO strategy based on the gradient of the economic function (ALVAREZ; ODLOAK, 2014; SANTANA; MARTINS; ODLOAK, 2020a).

- In Chapter 3, the invariant set for tracking was computed using the method proposed in Gilbert and Tan (1991). However, since a maximal admissible invariant set is not required, any method can be used to compute such invariant set (BLANCHINI, 1999), which can possibly enhance the applicability of the proposed controller. For instance, one can look into approaches suitable for designing invariant sets for large-scale systems (HENNET; CASTELAN, 2001).

- Robust stability can be addressed by extending the dual-mode MPC with OPOM for the case of multi-plant uncertainty (BADGWELL, 1997), polytopic uncertainty (KOTHARE; BALAKRISHNAN; MORARI, 1996) as well as considering bounded disturbances (CHISCI; ROSSITER; ZAPPA, 2001).

- A cooperative distributed MPC for dealing with unstable systems can be proposed by deploying the dual-mode framework with invariant set for tracking presented in Chapter 3. However, in order to avoid the design of a plantwide invariant set as employed in the cooperative distributed MPC presented in Ferramosca *et al.* (2013), one can explore the ideas of separable terminal cost and time-varying local invariant sets proposed in Conte *et al.* (2016).

- The robust cooperative distributed MPC algorithms proposed in this work consider the multi-plant uncertainty representation. The major limitation of this approach is the assumption that the true plant is represented by one of the models contained in the set Ω. Although this simplifies the problem and favors practical implementations, such an approach can be enhanced from a theoretical viewpoint. For instance, one may extend the proposed algorithms considering a polytopic description of the system (KOTHARE; BALAKRISHNAN; MORARI, 1996), which represents a more general form of uncertainty.

- A robustly stabilizing cooperative distributed gradient-based economic MPC can be proposed by combining the algorithms presented in Chapter 4 with a recently published cooperative distributed gradient-based economic MPC formulation (SANTANA *et al.*, 2022).

References

AL-GHERWI, W.; BUDMAN, H.; ELKAMEL, A. A robust distributed model predictive control algorithm. **Journal of Process Control**, v. 21, n. 8, p. 1127–1137, 2011.

AL-GHERWI, W.; BUDMAN, H.; ELKAMEL, A. A robust distributed model predictive control based on a dual-mode approach. **Computers & Chemical Engineering**, v. 50, p. 130–138, 2013.

ALVAREZ, L.; ODLOAK, D. Reduction of the QP-MPC cascade structure to a single layer MPC. **Journal of Process Control**, v. 24, n. 10, 2014.

ALVAREZ, L. A.; ODLOAK, D. Robust integration of real time optimization with linear model predictive control. **Computers & Chemical Engineering**, v. 34, n. 12, p. 1937–1944, 2010.

ANDERSSON, J. A. E.; GILLIS, J.; HORN, G.; RAWLINGS, J. B.; DIEHL, M. CasADi – A software framework for nonlinear optimization and optimal control. **Mathematical Programming Computation**, v. 11, n. 1, p. 1–36, 2019.

BADGWELL, T. A. Robust model predictive control of stable linear systems. **International Journal of Control**, v. 68, n. 4, p. 797–818, 1997.

BEMPORAD, A.; MORARI, M.; DUA, V.; PISTIKOPOULOS, E. N. The explicit linear quadratic regulator for constrained systems. **Automatica**, v. 38, n. 1, p. 3–20, 2002.

BLANCHINI, F. Set invariance in control. **Automatica**, v. 35, n. 11, p. 1747–1767, 1999.

CAMACHO, E. F.; BORDONS, C. **Model predictive control**. 2nd. ed. London: Springer-Verlag, 2007.

CARRAPIÇO, O.; SANTOS, M.; ZANIN, A.; ODLOAK, D. Application of the IHMPC to an industrial process system. **IFAC Proceedings Volumes**, v. 42, n. 11, p. 851–856, 2009.

CARRAPIÇO, O. L.; ODLOAK, D. A stable model predictive control for integrating processes. **Computers & Chemical Engineering**, v. 29, n. 5, p. 1089–1099, 2005.

CHISCI, L.; ROSSITER, J. A.; ZAPPA, G. Systems with persistent disturbances: predictive control with restricted constraints. **Automatica**, v. 37, n. 7, p. 1019–1028, 2001.

CHISCI, L.; ZAPPA, G. Dual mode predictive tracking of piecewise constant references for constrained linear systems. **International Journal of Control**, v. 76, n. 1, p. 61–72, 2003.

CLARKE, D.; MOHTADI, C.; TUFFS, P. S. Generalized Predictive Control - Part I. The Basic Algorithm. **Automatica**, v. 23, n. 2, p. 137–148, 1987.

CLARKE, D. W.; MOHTADI, C.; TUFFS, P. S. Generalized Predictive Control - Part II. Extensions and Interpretations. **Automatica**, v. 23, n. 6, p. 149–160, 1987.

CONTE, C.; JONES, C. N.; MORARI, M.; ZEILINGER, M. N. Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. **Automatica**, v. 69, p. 117–125, 2016.

CONTE, C.; ZEILINGER, M. N.; MORARI, M.; JONES, C. N. Robust distributed model predictive control of linear systems. *In*: **2013 European Control Conference (ECC)**. Zürich: IEEE, 2013. p. 2764–2769.

COSTA, T. V.; SENCIO, R. R.; OLIVEIRA-LOPES, L. C.; SILVA, F. V. Fault-tolerant control by means of moving horizon virtual actuators: Concepts and experimental investigation. **Control Engineering Practice**, v. 107, p. 104683, 2021.

CUTLER, C. R.; RAMAKER, B. L. **Dynamic matrix control – a computer control algorithm**. 1980.

DATTA, B. N. CHAPTER 8 - NUMERICAL SOLUTIONS AND CONDITIONING OF LYAPUNOV AND SYLVESTER EQUATIONS. *In*: DATTA, B. N. (ed.). **Numerical Methods for Linear Control Systems**. San Diego: Academic Press, 2004. p. 245–303.

DUGHMAN, S. S.; ROSSITER, J. A. A survey of guaranteeing feasibility and stability in MPC during target changes. **IFAC-PapersOnLine**, v. 28, n. 8, p. 813–818, 2015.

D'JORGE, A.; FERRAMOSCA, A.; GONZÁLEZ, A. H. A robust gradient-based MPC for integrating real time optimizer (RTO) with control. **Journal of Process Control**, v. 54, p. 65–80, 2017.

FERNANDES, D.; HAQUE, M. E.; PALANKI, S.; RIOS, S. G.; CHEN, D. DMC controller design for an integrated Allam cycle and air separation plant. **Computers & Chemical Engineering**, v. 141, p. 107019, 2020.

FERRAMOSCA, A.; GONZALEZ, A. H.; LIMON, D. Offset-free multi-model economic model predictive control for changing economic criterion. **Journal of Process Control**, v. 54, p. 1–13, 2017.

FERRAMOSCA, A.; LIMON, D.; ALVARADO, I.; ALAMO, T.; CAMACHO, E. F. MPC for tracking with optimal closed-loop performance. **Automatica**, v. 45, n. 8, p. 1975–1978, 2009.

FERRAMOSCA, A.; LIMON, D.; ALVARADO, I.; ALAMO, T.; CASTAÑO, F.; CAMACHO, E. F. Optimal mpc for tracking of constrained linear systems. **International Journal of Systems Science**, v. 42, n. 8, p. 1265–1276, 2011.

FERRAMOSCA, A.; LIMON, D.; ALVORADO, I.; CAMACHO, E. F. Cooperative distributed MPC for tracking. **Automatica**, v. 49, n. 4, p. 906–914, 2013.

FERRAMOSCA, A.; LIMON, D.; GONZÁLEZ, A. H.; ODLOAK, D.; CAMACHO, E. F. MPC for tracking zone regions. **Journal of Process Control**, v. 20, n. 4, p. 506–516, 2010.

FERREAU, H.; KIRCHES, C.; POTSCHKA, A.; BOCK, H.; DIEHL, M. qpOASES: A parametric active-set algorithm for quadratic programming. **Mathematical Programming Computation**, v. 6, n. 4, p. 327–363, 2014.

GARCIA, C. E.; MORARI, M. Internal Model Control. 1. A Unifying Review and Some New Results. **Industrial & Engineering Chemistry Process Design and Development**, v. 21, n. 2, p. 308–323, 1982.

GARCIA, C. E.; MORARI, M. Internal Model Control. 2. Design Procedure for Multivariable Systems. **Industrial & Engineering Chemistry Process Design and Development**, v. 24, n. 2, p. 472–484, 1985.

GARCIA, C. E.; MORARI, M. Internal Model Control. 3. Multivariable Control Law Computation and Tuning Guidelines. **Industrial & Engineering Chemistry Process Design and Development**, v. 24, n. 2, p. 484–494, 1985.

GARCIA, C. E.; MORSHEDI, A. Quadratic Programming Solution of Dynamic Matrix Control (QDMC). **Chemical Engineering Communications**, v. 46, n. 1-3, p. 73–87, 1986.

GARONE, E.; CAIRANO, S. D.; KOLMANOVSKY, I. Reference and command governors for systems with constraints: A survey on theory and applications. **Automatica**, v. 75, p. 306–328, 2017.

GERRISH, F.; WARD, A. J. B. Sylvester's matrix equation and Roth's removal rule. **The Mathematical Gazette**, v. 82, n. 495, p. 423–430, 1998.

GILBERT, E.; KOLMANOVSKY, I. Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. **Automatica**, v. 38, n. 12, p. 2063–2073, 2002.

GILBERT, E. G.; KOLMANOVSKY, I.; TAN, K. T. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. **International Journal of Robust and Nonlinear Control**, v. 5, n. 5, p. 487–504, 1995.

GILBERT, E. G.; TAN, K. T. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. **IEEE Transactions on Automatic Control**, v. 36, n. 9, p. 1008–1020, 1991.

GOLUB, G. H.; LOAN, C. F. V. **Matrix Computations**. 4th. ed. Baltimore: Johns Hopkins University Press, 2013.

GONZÁLEZ, A.; ADAM, E.; MARCHETTI, J. Conditions for offset elimination in state space receding horizon controllers: A tutorial analysis. **Chemical Engineering and Processing: Process Intensification**, v. 47, n. 12, p. 2184–2194, 2008.

GONZÁLEZ, A.; ODLOAK, D. Robust model predictive controller with output feedback and target tracking. **IET Control Theory & Applications**, v. 4, n. 8, p. 1377–1390, 2010.

GONZÁLEZ, A. H.; MARCHETTI, J. L.; ODLOAK, D. Extended Robust Model Predictive Control of Integrating Systems. **AIChE Journal**, v. 53, n. 504, p. 1758–1769, 2007.

GONZÁLEZ, A. H.; ODLOAK, D. A stable MPC with zone control. **Journal of Process Control**, v. 19, n. 1, p. 110–122, 2009.

GONZÁLEZ, A. H.; ODLOAK, D. Robust Model Predictive Control for Time Delayed Systems with Optimizing Targets and Zone Control. *In*: BARTOSZEWICZ, A. (ed.). **Robust control, theory and applications**. Rijeka: IntechOpen, 2011. chap. 15.

GONZÁLEZ, A. H.; PEREZ, J. M.; ODLOAK, D. Infinite horizon MPC with non-minimal state space feedback. **Journal of Process Control**, v. 19, n. 3, p. 473–481, 2009.

GOUVÊA, M. T.; ODLOAK, D. ROSSMPC: A new way of representing and analysing predictive controllers. **Chemical Engineering Research and Design**, v. 75, n. 7, p. 693–708, 1997.

HENNET, J. C.; CASTELAN, E. B. Invariant polyhedra and control of large scale systems. **IFAC Proceedings Volumes**, v. 34, n. 8, p. 563–568, 2001.

HIRIART-URRUTY, J. B.; LEMARÉCHAL, C. **Fundamentals of Convex Analysis**. Berlin Heidelberg: Springer-Verlag, 2001.

KANO, M.; OGAWA, M. The state of the art in chemical process control in Japan: Good practice and questionnaire survey. **Journal of Process Control**, v. 20, n. 9, p. 969–982, 2010.

KÖHLER, J.; MÜLLER, M. A.; ALLGÖWER, F. A nonlinear tracking model predictive control scheme for dynamic target signals. **Automatica**, v. 118, p. 109030, 2020.

KOTHARE, M. V.; BALAKRISHNAN, V.; MORARI, M. Robust constrained model predictive control using linear matrix inequalities. **Automatica**, v. 32, n. 10, p. 1361–1379, 1996.

LAHIRI, S. K. **Multivariable Predictive Control: Applications in Industry**. Hoboken, NJ, USA: John Wiley & Sons, 2017.

LI, H.; SHI, Y. Robust Distributed Model Predictive Control of Constrained Continuous-Time Nonlinear Systems: A Robustness Constraint Approach. **IEEE Transactions on Automatic Control**, v. 59, n. 6, p. 1673–1678, 2014.

LI, H.; SWARTZ, C. L. Dynamic real-time optimization of distributed MPC systems using rigorous closed-loop prediction. **Computers & Chemical Engineering**, v. 122, p. 356–371, 2019.

LI, S.; LIM, K. Y.; FISHER, D. G. A state space formulation for model predictive control. **AIChE Journal**, v. 35, n. 2, p. 241–249, 1989.

LIMON, D.; ALVARADO, I.; ALAMO, T.; CAMACHO, E. F. MPC for tracking piecewise constant references for constrained linear systems. **Automatica**, v. 44, n. 9, p. 2382–2387, 2008.

LIMON, D.; FERRAMOSCA, A.; ALVARADO, I.; ALAMO, T. Nonlinear mpc for tracking piece-wise constant reference signals. **IEEE Transactions on Automatic Control**, v. 63, n. 11, p. 3735–3750, 2018.

LIU, J.; de la PEÑA, D. M.; CHRISTOFIDES, P. D. Distributed Model Predictive Control of Nonlinear Process Systems. **AIChE Journal**, v. 55, n. 5, p. 1171–1184, 2009.

LIU, S.; MAO, Y.; LIU, J. Model-Predictive Control With Generalized Zone Tracking. **IEEE Transactions on Automatic Control**, v. 64, n. 11, p. 4698–4704, 2019.

LOBO, M. S.; VANDENBERGHE, L.; BOYD, S.; LEBRET, H. Applications of second-order cone programming. **Linear Algebra and its Applications**, v. 284, n. 1, p. 193–228, 1998.

MACIEJOWSKI, J. M. **Predictive control: with constraints**. London: Prentice Hall, 2002.

MARQUIS, P.; BROUSTAIL, J. SMOC, a bridge between state space and model predictive controllers: Application to the automation of a hydrotreating unit. **IFAC Proceedings Volumes**, v. 21, n. 4, p. 37–45, 1988. IFAC Workshop on Model Based Process Control, Atlanta, GA, USA, 13-14 June.

MARTIN, P. A.; ODLOAK, D.; KASSAB, F. Robust model predictive control of a pilot plant distillation column. **Control Engineering Practice**, v. 21, n. 3, p. 231–241, 2013.

MARTIN, P. A.; ZANIN, A. C.; ODLOAK, D. Integrating real time optimization and model predictive control of a crude distillation unit. **Brazilian Journal of Chemical Engineering**, v. 36, p. 1205–1222, 2019.

MARTINS, M. A. F. **Robust model predictive control of integrating and unstable time delay processes**. 2014. Thesis (PhD in Engineering) — Polytechnic School of the University of São Paulo, São Paulo, 2014.

MARTINS, M. A. F.; ODLOAK, D. A robustly stabilizing model predictive control strategy of stable and unstable processes. **Automatica**, v. 67, p. 132–143, 2016.

MARTINS, M. A. F.; YAMASHITA, A. S.; SANTORO, B. F.; ODLOAK, D. Robust model predictive control of integrating time delay processes. **Journal of Process Control**, v. 23, n. 7, p. 917–932, 2013.

MARTINS, M. A. F.; ZANIN, A. C.; ODLOAK, D. Robust model predictive control of an industrial partial combustion fluidized-bed catalytic cracking converter. **Chemical Engineering Research and Design**, v. 92, n. 5, p. 917–930, 2014.

MASERO, E.; MAESTRE, J. M.; MEMBER, S.; FERRAMOSCA, A.; FRANCISCO, M.; CAMACHO, E. F. Robust Coalitional Model Predictive Control With Predicted Topology Transitions. **IEEE Transactions on Control of Network Systems**, v. 8, n. 4, p. 1869–1880, 2021.

MAYNE, D.; FALUGI, P. Generalized Stabilizing Conditions for Model Predictive Control. **Journal of Optimization Theory and Applications**, v. 169, n. 3, p. 719–734, 2016.

MAYNE, D.; RAWLINGS, J.; RAO, C.; SCOKAERT, P. Constrained model predictive control: Stability and optimality. **Automatica**, v. 36, n. 6, p. 789–814, 2000.

MAYNE, D. Q.; FALUGI, P. Stabilizing conditions for model predictive control. **International Journal of Robust and Nonlinear Control**, v. 29, n. 4, p. 894–903, 2019.

MORARI, M.; ZAFIRIOU, E. **Robust Process Control**. New Jersey: Prentice-Hall, Inc., 1989.

MORSHEDI, A.; CUTLER, C. R.; SKROVANEK, T. A. Optimal Solution of Dynamic Matrix Control with Linear Programing Techniques (LDMC). *In*: **1985 American Control Conference**. Boston, MA, USA: IEEE, 1985. p. 199–208.

MUSKE, K. R.; BADGWELL, T. A. Disturbance modeling for offset-free linear model predictive control. **Journal of Process Control**, v. 12, n. 5, p. 617–632, 2002.

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

NAGAR, S. K.; SINGH, S. K. An algorithmic approach for system decomposition and balanced realized model reduction. **Journal of the Franklin Institute**, v. 341, n. 7, p. 615–630, 2004.

OBERDIECK, R.; DIANGELAKIS, N. A.; NASCU, I.; PAPATHANASIOU, M. M.; SUN, M.; AVRAAMIDOU, S.; PISTIKOPOULOS, E. N. On multi-parametric programming and its applications in process systems engineering. **Chemical Engineering Research and Design**, v. 116, p. 61–82, 2016.

ODLOAK, D. Extended robust model predictive control. **AIChE Journal**, v. 50, n. 8, p. 1824–1836, 2004.

PANNOCCHIA, G. Offset-free tracking MPC: A tutorial review and comparison of different formulations. *In*: **2015 European Control Conference (ECC)**. Linz, Austria: IEEE, 2015. p. 527–532.

PANNOCCHIA, G.; RAWLINGS, J. B. Disturbance Models for Offset-Free Model-Predictive Control. **AIChE Journal**, v. 49, n. 2, p. 426–437, 2003.

PAULSON, J. A.; SANTOS, T. L.; MESBAH, A. Mixed stochastic-deterministic tube MPC for offset-free tracking in the presence of plant-model mismatch. **Journal of Process Control**, v. 83, p. 102–120, 2019.

PEREZ, J.; ODLOAK, D. Robust MPC with Output Feedback and Realigned Model. **IFAC Proceedings Volumes**, v. 39, n. 2, p. 317–322, 2006. 6th IFAC Symposium on Advanced Control of Chemical Processes.

PEREZ, J. M.; ODLOAK, D.; LIMA, E. L. Robust MPC with output feedback of integrating systems. **Journal of Control Science and Engineering**, v. 2012, 2012.

PEREZ, J. M.; ODLOAK, D.; LIMA, E. L. Multi-model mpc with output feedback. **Brazilian Journal of Chemical Engineering**, v. 31, n. 1, p. 131–144, 2014.

PORFÍRIO, C. R.; ALMEIDA NETO, E.; ODLOAK, D. Multi-model predictive control of an industrial C3/C4 splitter. **Control Engineering Practice**, v. 11, n. 7, p. 765–779, 2003.

QIN, S.; BADGWELL, T. A. A survey of industrial model predictive control technology. **Control Engineering Practice**, v. 11, n. 7, p. 733–764, 2003.

RALHAN, S.; BADGWELL, T. A. Robust control of stable linear systems with continuous uncertainty. **Computers & Chemical Engineering**, v. 24, n. 11, p. 2533–2544, 2000.

RAO, C. V.; RAWLINGS, J. B. Steady States and Constraints in Model Predictive Control. **AIChE Journal**, v. 45, n. 6, p. 1266–1278, 1999.

RAWLINGS, J. Tutorial overview of model predictive control. **IEEE Control Systems Magazine**, v. 20, n. 3, p. 38–52, 2000.

RAWLINGS, J. B.; MAYNE, D. Q.; DIEHL, M. **Model Predictive Control: Theory, Computation, and Design**. 2nd. ed. Santa Barbara, California, USA: Nob Hill Publishing Madison, 2017.

RAWLINGS, J. B.; MUSKE, K. R. The stability of constrained receding horizon control. **IEEE Transactions on Automatic Control**, v. 38, n. 10, p. 1512–1516, 1993.

RAZAVINASAB, Z.; FARSANGI, M. M.; BARKHORDARI, M. Robust output feedback distributed model predictive control of networked systems with communication delays in the presence of disturbance. **ISA Transactions**, v. 80, p. 12–21, 2018.

RAZZANELLI, M.; PANNOCCHIA, G. Parsimonious cooperative distributed MPC algorithms for offset-free tracking. **Journal of Process Control**, v. 60, p. 1–13, 2017.

RIBEIRO, J. B. A.; GESSER, R. D. S.; LIMA, D. M.; NORMEY-RICO, J. E. Controle preditivo robusto baseado em tubos para modelos de entrada e saída. *In*: **Anais do XXIII Congresso Brasileiro de Automática**. Virtual (UFSM/PUCRS/UNISINOS): SBA, 2020.

RICHALET, J.; RAULT, A.; TESTUD, J.; PAPON, J. Model predictive heuristic control. **Automatica**, v. 14, n. 5, p. 413–428, 1978.

RICHARDS, A.; HOW, J. Robust distributed model predictive control. **International Journal of Control**, v. 80, n. 9, p. 1517–1531, 2007.

RODRIGUES, M. A.; ODLOAK, D. Output feedback MPC with guaranteed robust stability. **Journal of Process Control**, v. 10, n. 6, p. 557–572, 2000.

RODRIGUES, M. A.; ODLOAK, D. An infinite horizon model predictive control for stable and integrating processes. **Computers & Chemical Engineering**, v. 27, n. 8-9, p. 1113–1128, 2003.

RODRIGUES, M. A.; ODLOAK, D. MPC for stable linear systems with model uncertainty. **Automatica**, v. 39, n. 4, p. 569–583, 2003.

ROSSITER, J.; KOUVARITAKIS, B. Constrained stable generalised predictive control. **IEE Proceedings-D Control Theory and Applications**, v. 140, n. 4, p. 243–254, 1993. Oxford University Research Archive.

ROSSITER, J. A. **Model-Based Predictive Control: A Practical Approach**. Boca Raton, Florida, USA: CRC Press, 2004.

ROSSITER, J. A.; KOUVARITAKIS, B.; GOSSNER, J. R. Guaranteeing feasibility in constrained stable generalised predictive control. **IEE Proceedings - Control Theory and Applications**, v. 143, n. 5, p. 463–469, 1996.

ROUHANI, R.; MEHRA, R. K. Model algorithmic control (MAC): basic theoretical properties. **Automatica**, v. 18, n. 4, p. 401–414, 1982.

RUSSO, L. P.; BEQUETTE, B. W. State-space versus input/output representations for cascade control of unstable systems. **Industrial & Engineering Chemistry Research**, v. 36, n. 6, p. 2271–2278, 1997.

SANTANA, D. D. **Economic and distributed model predictive control for non-stable processes: stability, feasibility, and integration**. 2020. Thesis (PhD in Industrial Engineering) — Universidade Federal da Bahia, Salvador, 2020.

SANTANA, D. D.; MARTINS, M. A.; ODLOAK, D. One-layer gradient-based MPC + RTO strategy for unstable processes: a case study of a CSTR system. **Brazilian Journal of Chemical Engineering**, v. 37, n. 1, p. 173–188, 2020.

SANTANA, D. D.; MARTINS, M. A. F.; ODLOAK, D. An efficient cooperative-distributed model predictive controller with stability and feasibility guarantees for constrained linear systems. **Systems & Control Letters**, v. 141, 2020.

SANTANA, D. D.; ODLOAK, D.; SANTOS, T. L.; MARTINS, M. A. A stabilizing cooperative-distributed gradient-based economic model predictive control strategy for constrained linear systems. **Journal of Process Control**, v. 112, p. 36–48, 2022.

SANTORO, B. F.; ODLOAK, D. Closed-loop stable model predictive control of integrating systems with dead time. **Journal of Process Control**, v. 22, n. 7, p. 1209–1218, 2012.

SARAPKA, A. S. **MPC distribuído e cooperativo com estabilidade garantida associado a camada de otimização (RTO) e com múltiplas escalas temporais**. 2020. Dissertation (Master of Science) — Polytechnic School of the University of São Paulo, São Paulo, 2020.

SARAPKA, A. S.; MARTINS, M. A.; ODLOAK, D. Stable distributed MPC with zone control and input targets. **Computers & Chemical Engineering**, v. 155, p. 107507, 2021.

SCATTOLINI, R. Architectures for distributed and hierarchical Model Predictive Control - A review. **Journal of Process Control**, v. 19, n. 5, p. 723–731, 2009.

SCOKAERT, P. O.; RAWLINGS, J. B. Feasibility issues in linear model predictive control. **AIChE Journal**, v. 45, n. 8, p. 1649–1659, 1999.

SCOKAERT, P. O. M.; RAWLINGS, J. B. Constrained Linear Quadratic Regulation. **IEEE Transactions on Automatic Control**, v. 43, n. 8, p. 1163–1169, 1998.

SENCIO, R. R.; ODLOAK, D. An Infinite Horizon Model Predictive Control for Stable, Integrating and Unstable Systems. *In*: **Anais do XXII Congresso Brasileiro de Automática**. João Pessoa, Brazil: SBA, 2018.

SENCIO, R. R.; ODLOAK, D. A general infinite horizon mpc formulation for stable, integrating and unstable systems. **Blucher Chemical Engineering Proceedings**, v. 1, n. 5, p. 1253 – 1256, 2018. XXII Congresso Brasileiro de Engenharia Química.

SENCIO, R. R.; ODLOAK, D. Robust cooperative distributed MPC: A multi-model approach. **Journal of Process Control**, v. 117, p. 65–77, 2022.

SHALMANI, R. A.; RAHMANI, M.; BIGDELI, N. Nash-based robust distributed model predictive control for large-scale systems. **Journal of Process Control**, v. 88, p. 43–53, 2020.

SILVA, B. P.; SANTANA, B. A.; SANTOS, T. L.; MARTINS, M. A. An implementable stabilizing model predictive controller applied to a rotary flexible link: An experimental case study. **Control Engineering Practice**, v. 99, p. 104396, 2020.

SIMON, D.; LOFBERG, J.; GLAD, T. Reference tracking MPC using dynamic terminal set transformation. **IEEE Transactions on Automatic Control**, v. 59, n. 10, p. 2790–2795, 2014.

SKOGESTAD, S.; POSTLETHWAITE, I. **Multivariable Feedback Control: Analysis and Design**. 2nd. ed. Chichester, England: John Wiley & Sons, 2005.

STEWART, B. T.; VENKAT, A. N.; RAWLINGS, J. B.; WRIGHT, S. J.; PANNOCCHIA, G. Cooperative distributed model predictive control. **Systems & Control Letters**, v. 59, n. 8, p. 460–469, 2010.

TRODDEN, P.; RICHARDS, A. Cooperative distributed MPC of linear systems with coupled constraints. **Automatica**, v. 49, n. 2, p. 479–487, 2013.

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

WANG, L. A Tutorial on Model Predictive Control: Using a Linear Velocity-Form Model. **Developments in Chemical Engineering and Mineral Processing**, v. 12, n. 5-6, p. 573–614, 2004.

WANG, L.; YOUNG, P. C. An improved structure for model predictive control using non-minimal state space realisation. **Journal of Process Control**, v. 16, n. 4, p. 355–371, 2006.

WANG, Y.; MANZIE, C. Robust Cooperative Distributed Model Predictive Control based on Set-membership Approach. *In*: **2020 Australian and New Zealand Control Conference (ANZCC)**. Gold Coast: IEEE, 2020. p. 166–171.

WHEAT, P. **Multivariable model predictive control of a pilot plant using Honeywell profit suite**. 2018. Thesis (Engineering Honours) — Murdoch University, Perth, 2018.

YING, C. M.; VOORAKARANAM, S.; JOSEPH, B. Performance and Stability Analysis of LP-MPC and QP-MPC Cascade Control Systems. **AIChE Journal**, v. 45, n. 7, p. 1521–1534, 1999.

YOUNG, P.; BEHZADI, M. A.; WANG, C. L.; CHOTAI, A. Direct digital and adaptive control by input-output state variable feedback pole assignment. **International Journal of Control**, v. 46, n. 6, p. 1867–1881, 1987.

YOUSFI, C.; TOURNIER, R. Steady State Optimization Inside Model Predictive Control. *In*: **1991 American Control Conference**. Boston, MA, USA: IEEE, 1991. p. 1866–1870.

ZEILINGER, M. N.; MORARI, M.; JONES, C. N. Soft Constrained Model Predictive Control With Robust Stability Guarantees. **IEEE Transactions on Automatic Control**, v. 59, n. 5, p. 1190–1202, 2014.

ZHANG, R.; XUE, A.; WANG, S.; REN, Z. An improved model predictive control approach based on extended non-minimal state space formulation. **Journal of Process Control**, v. 21, n. 8, p. 1183–1192, 2011.

Appendix A

System decomposition into stable, integrating and unstable subsystems

Consider the following discrete-time linear system with n_{st} stable, n_{in} integrating and n_{un} unstable poles, n_u inputs and n_y outputs:

$$x^+ = Ax + Bu \tag{A.1}$$

$$y = Cx \tag{A.2}$$

A real Schur decomposition of A is given as follows:

$$U^T A U = \begin{bmatrix} \tilde{A}_{st} & \tilde{A}_{c1} & \tilde{A}_{c2} \\ 0 & \tilde{A}_{in} & \tilde{A}_{c3} \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} \tag{A.3}$$

in which U is an orthogonal matrix ($U^T U = UU^T = I$), and, without loss of generality, the transformed matrix is arranged such that the submatrices $\tilde{A}_{st} \in \mathbb{R}^{n_{st} \times n_{st}}$, $\tilde{A}_{in} \in \mathbb{R}^{n_{in} \times n_{in}}$, $\tilde{A}_{un} \in \mathbb{R}^{n_{un} \times n_{un}}$ contain only stable, integrating and unstable eigenvalues, respectively.

A block diagonal matrix can be obtained by applying the Roth's removal rule (GERRISH; WARD, 1998; NAGAR; SINGH, 2004):

$$W^{-1} \begin{bmatrix} \tilde{A}_{st} & \tilde{A}_{c1} & \tilde{A}_{c2} \\ 0 & \tilde{A}_{in} & \tilde{A}_{c3} \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} W = \begin{bmatrix} \tilde{A}_{st} & 0 & 0 \\ 0 & \tilde{A}_{in} & 0 \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} \tag{A.4}$$

with $W = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix}$.

To determine the submatrices of W, let us rewrite (A.4) as follows:

$$\begin{bmatrix} \tilde{A}_{st} & \tilde{A}_{c1} & \tilde{A}_{c2} \\ 0 & \tilde{A}_{in} & \tilde{A}_{c3} \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} W = W \begin{bmatrix} \tilde{A}_{st} & 0 & 0 \\ 0 & \tilde{A}_{in} & 0 \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} \tag{A.5}$$

Then, the left-hand side of (A.5) is given as follows:

$$\begin{bmatrix} \tilde{A}_{st} & \tilde{A}_{c1} & \tilde{A}_{c2} \\ 0 & \tilde{A}_{in} & \tilde{A}_{c3} \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} W =$$

$$\begin{bmatrix} \tilde{A}_{st}W_{11} + \tilde{A}_{c1}W_{21} + \tilde{A}_{c2}W_{31} & \tilde{A}_{st}W_{12} + \tilde{A}_{c1}W_{22} + \tilde{A}_{c2}W_{32} & \tilde{A}_{st}W_{13} + \tilde{A}_{c1}W_{23} + \tilde{A}_{c2}W_{33} \\ \tilde{A}_{in}W_{21} + \tilde{A}_{c3}W_{31} & \tilde{A}_{in}W_{22} + \tilde{A}_{c3}W_{32} & \tilde{A}_{in}W_{23} + \tilde{A}_{c3}W_{33} \\ \tilde{A}_{un}W_{31} & \tilde{A}_{un}W_{32} & \tilde{A}_{un}W_{33} \end{bmatrix} \tag{A.6}$$

and the right-hand side is:

$$W \begin{bmatrix} \tilde{A}_{st} & 0 & 0 \\ 0 & \tilde{A}_{in} & 0 \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} = \begin{bmatrix} W_{11}\tilde{A}_{st} & W_{12}\tilde{A}_{in} & W_{13}\tilde{A}_{un} \\ W_{21}\tilde{A}_{st} & W_{22}\tilde{A}_{in} & W_{23}\tilde{A}_{un} \\ W_{31}\tilde{A}_{st} & W_{32}\tilde{A}_{in} & W_{33}\tilde{A}_{un} \end{bmatrix} \tag{A.7}$$

From the right-hand side of (A.6) and (A.7), we can write the following equations:

$$\tilde{A}_{st}W_{11} + \tilde{A}_{c1}W_{21} + \tilde{A}_{c2}W_{31} = W_{11}\tilde{A}_{st} \tag{A.8}$$

$$\tilde{A}_{st}W_{12} + \tilde{A}_{c1}W_{22} + \tilde{A}_{c2}W_{32} = W_{12}\tilde{A}_{in} \tag{A.9}$$

$$\tilde{A}_{st}W_{13} + \tilde{A}_{c1}W_{23} + \tilde{A}_{c2}W_{33} = W_{13}\tilde{A}_{un} \tag{A.10}$$

$$\tilde{A}_{in}W_{21} + \tilde{A}_{c3}W_{31} = W_{21}\tilde{A}_{st} \tag{A.11}$$

$$\tilde{A}_{in}W_{22} + \tilde{A}_{c3}W_{32} = W_{22}\tilde{A}_{in} \tag{A.12}$$

$$\tilde{A}_{in}W_{23} + \tilde{A}_{c3}W_{33} = W_{23}\tilde{A}_{un} \tag{A.13}$$

$$\tilde{A}_{un}W_{31} = W_{31}\tilde{A}_{st} \tag{A.14}$$

$$\tilde{A}_{un}W_{32} = W_{32}\tilde{A}_{in} \tag{A.15}$$

$$\tilde{A}_{un}W_{33} = W_{33}\tilde{A}_{un} \tag{A.16}$$

Note that these 9 equations (A.8)-(A.16) can be used to obtain the submatrices of W. From (A.14), we have that:

$$W_{31} = 0 \tag{A.17}$$

From (A.15):

$$W_{32} = 0 \tag{A.18}$$

From (A.16):

$$W_{33} = I \tag{A.19}$$

From (A.11) and (A.17):

$$W_{21} = 0 \tag{A.20}$$

From (A.12) and (A.18):

$$W_{22} = I \tag{A.21}$$

From (A.13) and (A.19):

$$\tilde{A}_{in} W_{23} - W_{23} \tilde{A}_{un} + \tilde{A}_{c3} = 0 \tag{A.22}$$

From (A.8), (A.17) and (A.20):

$$W_{11} = I \tag{A.23}$$

From (A.9), (A.18) and (A.21):

$$\tilde{A}_{st} W_{12} - W_{12} \tilde{A}_{in} + \tilde{A}_{c1} = 0 \tag{A.24}$$

From (A.10), (A.19) and (A.22):

$$\tilde{A}_{st} W_{13} - W_{13} \tilde{A}_{un} + \tilde{A}_{c1} W_{23} + \tilde{A}_{c2} = 0 \tag{A.25}$$

in which W_{23} can be computed from (A.22).

Equations (A.23), (A.24) and (A.25) are Sylvester equations since they have the form of $MX + XN = Q$, in which matrices M, N and Q are of dimensions $m \times m$, $n \times n$ and $m \times n$, respectively. A unique solution X exists if and only if the spectra of M and $-N$ are disjoint (DATTA, 2004).

Since $\tilde{A}_{st}$, $\tilde{A}_{in}$ and $\tilde{A}_{un}$ have no eigenvalues in common, equations (A.22), (A.24) and (A.25) can be solved to obtain W_{23}, W_{12} and W_{13}, respectively. Therefore, matrix W has the following form:

$$W = \begin{bmatrix} I & W_{12} & W_{13} \\ 0 & I & W_{23} \\ 0 & 0 & I \end{bmatrix} \tag{A.26}$$

Now, let us compute $W^{-1} = \tilde{W}$ using the fact that $W\tilde{W} = I$. In terms of submatrices, we have that:

$$\begin{bmatrix} I & W_{12} & W_{13} \\ 0 & I & W_{23} \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \tilde{W}_{11} & \tilde{W}_{12} & \tilde{W}_{13} \\ \tilde{W}_{21} & \tilde{W}_{22} & \tilde{W}_{23} \\ \tilde{W}_{31} & \tilde{W}_{32} & \tilde{W}_{33} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \tag{A.27}$$

which is equivalent to

$$\begin{bmatrix} \tilde{W}_{11} + W_{12}\tilde{W}_{21} + W_{13}\tilde{W}_{31} & \tilde{W}_{12} + W_{12}\tilde{W}_{22} + W_{13}\tilde{W}_{32} & \tilde{W}_{13} + W_{12}\tilde{W}_{23} + W_{13}\tilde{W}_{33} \\ \tilde{W}_{21} + W_{23}\tilde{W}_{31} & \tilde{W}_{22} + W_{23}\tilde{W}_{32} & \tilde{W}_{23} + W_{23}\tilde{W}_{33} \\ \tilde{W}_{31} & \tilde{W}_{32} & \tilde{W}_{33} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \tag{A.28}$$

From (A.28), the following equations can be written:

$$\tilde{W}_{11} + W_{12}\tilde{W}_{21} + W_{13}\tilde{W}_{31} = I \tag{A.29}$$
$$\tilde{W}_{12} + W_{12}\tilde{W}_{22} + W_{13}\tilde{W}_{32} = 0 \tag{A.30}$$
$$\tilde{W}_{13} + W_{12}\tilde{W}_{23} + W_{13}\tilde{W}_{33} = 0 \tag{A.31}$$
$$\tilde{W}_{21} + W_{23}\tilde{W}_{31} = 0 \tag{A.32}$$
$$\tilde{W}_{22} + W_{23}\tilde{W}_{32} = I \tag{A.33}$$
$$\tilde{W}_{23} + W_{23}\tilde{W}_{33} = 0 \tag{A.34}$$
$$\tilde{W}_{31} = 0 \tag{A.35}$$
$$\tilde{W}_{32} = 0 \tag{A.36}$$
$$\tilde{W}_{33} = I \tag{A.37}$$

Then, from (A.32) and (A.35):

$$\tilde{W}_{21} = 0 \tag{A.38}$$

From (A.33) and (A.36):

$$\tilde{W}_{22} = 0 \tag{A.39}$$

From (A.34) and (A.37):

$$\tilde{W}_{23} = -W_{23} \tag{A.40}$$

From (A.29), (A.35) and (A.38):

$$\tilde{W}_{11} = I \tag{A.41}$$

From (A.30), (A.36) and (A.39):

$$\tilde{W}_{12} = -W_{12} \tag{A.42}$$

From (A.31), (A.37) and (A.40):

$$\tilde{W}_{13} = W_{12}W_{23} - W_{13} \tag{A.43}$$

Therefore, we have that $W^{-1} = \tilde{W}$ is given as follows:

$$W^{-1} = \begin{bmatrix} I & -W_{12} & W_{12}W_{23} - W_{13} \\ 0 & I & -W_{23} \\ 0 & 0 & I \end{bmatrix} \tag{A.44}$$

Note that no matrix inversion is required since W^{-1} was computed analytically.

Now, let $T = UW$ and $T^{-1} = W^{-1}U^T$. Then, the following similarity transformations decompose the original system (A.1)-(A.2) into stable, integrating and unstable subsystems:

$$T^{-1}x = \begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \\ \tilde{x}_{un} \end{bmatrix}, \quad T^{-1}AT = \begin{bmatrix} \tilde{A}_{st} & 0 & 0 \\ 0 & \tilde{A}_{in} & 0 \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix}, \quad T^{-1}B = \begin{bmatrix} \tilde{B}_{st} \\ \tilde{B}_{in} \\ \tilde{B}_{un} \end{bmatrix}$$

$$CT = \begin{bmatrix} \tilde{C}_{st} & \tilde{C}_{in} & \tilde{C}_{un} \end{bmatrix}$$

This results in the following decomposed system:

$$\begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \\ \tilde{x}_{un} \end{bmatrix}^{+} = \begin{bmatrix} \tilde{A}_{st} & 0 & 0 \\ 0 & \tilde{A}_{in} & 0 \\ 0 & 0 & \tilde{A}_{un} \end{bmatrix} \begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \\ \tilde{x}_{un} \end{bmatrix} + \begin{bmatrix} \tilde{B}_{st} \\ \tilde{B}_{in} \\ \tilde{B}_{un} \end{bmatrix} u$$

$$y = \begin{bmatrix} \tilde{C}_{st} & \tilde{C}_{in} & \tilde{C}_{un} \end{bmatrix} \begin{bmatrix} \tilde{x}_{st} \\ \tilde{x}_{in} \\ \tilde{x}_{un} \end{bmatrix}$$

in which $\tilde{x}_{st} \in \mathbb{R}^{n_{st}}$, $\tilde{x}_{in} \in \mathbb{R}^{n_{in}}$, $\tilde{x}_{un} \in \mathbb{R}^{n_{un}}$, $\tilde{A}_{st} \in \mathbb{R}^{n_{st} \times n_{st}}$, $\tilde{A}_{in} \in \mathbb{R}^{n_{in} \times n_{in}}$, $\tilde{A}_{un} \in \mathbb{R}^{n_{un} \times n_{un}}$, $\tilde{B}_{st} \in \mathbb{R}^{n_{st} \times n_u}$, $\tilde{B}_{in} \in \mathbb{R}^{n_{in} \times n_u}$, $\tilde{B}_{un} \in \mathbb{R}^{n_{un} \times n_u}$, $\tilde{C}_{st} \in \mathbb{R}^{n_y \times n_{st}}$, $\tilde{C}_{in} \in \mathbb{R}^{n_y \times n_{in}}$ and $\tilde{C}_{un} \in \mathbb{R}^{n_y \times n_{un}}$.