Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia Química
Alexandre Sanioto Sarapka

MPC distribuído e cooperativo com
estabilidade garantida associado à camada de
otimização (RTO) e com múltiplas escalas
temporais

São Paulo
2020

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia Química
Alexandre Sanioto Sarapka

MPC distribuído e cooperativo com
estabilidade garantida associado à camada de
otimização (RTO) e com múltiplas escalas
temporais

Versão Corrigida

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para a obten-
ção do título de Mestre em Ciências.
Área de Concentração: Engenharia Química
Orientador: Prof. Dr. Darci Odloak

São Paulo
2020

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, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Sarapka, Alexandre Sanioto

MPC distribuído e cooperativo com estabilidade garantida associado a camada de otimização (RTO) e com múltiplas escalas temporais / A. S. Sarapka, D. Odloak (orientador) -- versão corr. -- São Paulo, 2020.
152 p.

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

1.Controle preditivo baseado em modelo 2.MPC 3.MPC distribuído e cooperativo 4.RTO 5.Múltiplas escalas temporais I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Química II.t. III.Odloak (orientador), Darci

Agradecimentos

Ao Prof. Dr. Darci Odloak, pela orientação, compreensão, apoio e paciência em todos os momentos.

Ao Prof. Dr. Luciano Gonçalves Ribeiro, pelo incentivo em iniciar a jornada.

Ao Prof. Dr. José Luís de Paiva, pelas longas e inspiradoras conversas. Especialmente nos momentos difíceis.

À Paula Watanabe Martins, minha mulher, pelo apoio e carinho.

A toda a minha família, desde o princípio.

Carrego comigo
há dezenas de anos
há centenas de anos
o pequeno embrulho.
[...]
Não sei o que seja.
Eu não a escolhi.
Jamais a fitei.
Mas levo uma coisa.

Não estou vazio,
não estou sozinho,
pois anda comigo
algo indescritível.

Carlos Drummond de Andrade,
A Rosa do Povo

Resumo

SARAPKA, Alexandre Sanioto. MPC distribuído e cooperativo com estabilidade garantida associado à camada de otimização (RTO) e com múltiplas escalas temporais. Dissertação (Mestrado em Engenharia Química) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2020.

O “controle preditivo baseado em modelo” (em inglês: *Model Predictive Control*) ou ainda “MPC” ganhou protagonismo no controle de processos da indústria química por sua superioridade em relação a outras tecnologias de controle. Seu funcionamento baseia-se em estimar e minimizar, a partir de um modelo de funcionamento da planta, a diferença entre um estado desejado e um estado previsto - calculado a partir deste modelo e das entradas da planta. Além disso, o MPC permite levar em conta restrições, como os limites físicos da planta e de atuadores. Em suma, o MPC funciona como uma minimização de função com restrições. No presente trabalho, desenvolveu-se o chamado “MPC Distribuído e Cooperativo” (em inglês: *Distributed and Cooperative Model Predictive Control* (DCMPC)). Tal MPC resolve o problema de descentralizar o controle da planta com confiabilidade. A necessidade de descentralização, por sua vez, surge a partir do alto custo e alta complexidade dos sistemas atuais, bem como da possibilidade de se incluir no pacote de controle tecnologias complementares, as quais seriam impraticáveis em sistemas centralizados de grande porte. Foram desenvolvidos diferentes tipos de MPCs: o primeiro é capaz de atuar em sistemas quadrados com alvos nas variáveis controladas, o segundo atua em uma estrutura com camadas superiores de otimização (RTO), o terceiro é uma variação que permite lidar com tempo morto, finalmente, o quarto é capaz de lidar com sistemas com dinâmicas temporais distintas, operando em múltiplas escalas de tempo. Para ilustrar o desempenho dos controladores, foram feitas simulações de casos teóricos e de um caso industrial: o conjunto de fracionadoras de petróleo cru da refinaria de Capuava, SP - Petrobras. Em todas as simulações, os controladores foram capazes de controlar os sistemas adequadamente. No caso industrial, o controlador desenvolvido levou 25.30ms para resolver o problema de otimização, enquanto que o controlador centralizado levou 63.14ms para resolver o mesmo problema.

Palavras-chave: Controle preditivo baseado em modelo (MPC). MPC distribuído e cooperativo. RTO. Múltiplas escalas temporais

Abstract

SARAPKA, Alexandre Sanioto. Distributed and cooperative MPC with guaranteed stability, RTO integration and multiple time scales. Dissertação (Mestrado em Engenharia Química) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2020.

Model Predictive Control (MPC) has obtained protagonism in industrial process control due to its superiority compared with other control technologies. It works by estimating and minimizing the difference between desired and forecasted states, based on a state model. Besides, MPC allows to account for actuators and plant constraints. To sum up, MPC usually works as a quadratic programming optimization problem with constraints. The current work addressed the *Distributed and Cooperative Model Predictive Control* (DCMPC). Such system targets the decentralization problem without control or reliability losses. The need for decentralization, on the other hand arises from the high costs and complexity in large scale facilities, as well as the possibility to include several other technologies in the control package - which would be impractical in large centralized systems. Moreover, it was developed in this work several control packages: the first one deals with output set-points; the second one with input targets and optimization integration; the third one is an adaptation to deal with time delayed systems and finally, the forth one deals with systems with different dynamics, which require multiple time scales. To illustrate the proposed control approach, simple as well as complex (*Crude Distillation Unit* - Capuava, Brazil - Petrobras) systems were simulated.

Keywords: *Model Predictive Control*; MPC; *Distributed and Cooperative Model Predictive Control*; RTO; Multiple time scales

Lista de Ilustrações

Figura 1	Estrutura de controle: Integração RTO/MPC	50
Figura 2	Comparação da arquitetura MPC tradicional vs. cascadeado	75
Figura 3	Simulação: sistema 2×2 - MPC centralizado vs. distribuído - variação de p	88
Figura 4	Custo: sistema 2×2 - MPC centralizado vs. distribuído - variação de p	89
Figura 5	Simulação: sistema 2×2 - MPC centralizado vs. distribuído - variação de $\mathbf{w}$	90
Figura 6	Custo: sistema 2×2 - MPC centralizado vs. distribuído - variação de $\mathbf{w}$	90
Figura 7	Simulação: sistema 3×3 - MPC centralizado vs. distribuído	92
Figura 8	Custos: sistema 3×3 - MPC centralizado vs. distribuído	93
Figura 9	Estrutura de controle: Integração RTO/DCMPC	95
Figura 10	Simulação: saídas para um sistema 3×2 com RTO	106
Figura 11	Simulação: entradas para um sistema 3×2 com RTO	107
Figura 12	Função objetivo para um sistema 3×2 com RTO	107
Figura 13	Simulação: saídas para um sistema 3×2 com tempo morto	112
Figura 14	Simulação: entradas para um sistema 3×2 com tempo morto	113
Figura 15	Função objetivo para um sistema 3×2 com tempo morto	113
Figura 16	Simulação: dois subsistemas com múltiplas escalas temporais	120
Figura 17	Função objetivo: dois subsistemas com múltiplas escalas temporais	120
Figura 18	Simulação: três subsistemas com múltiplas escalas temporais	121
Figura 19	Função objetivo: três subsistemas com múltiplas escalas temporais .	121
Figura 20	Esquema da unidade de destilação de óleo cru da unidade de refino da Petrobras de Capuava-SP	122
Figura 21	CDU e sua divisão em três subsistemas	124
Figura 22	Simulação: caso industrial (CDU) - entrada u_1 (Fluxo de cru)	126
Figura 23	Simulação: caso industrial (CDU) - entrada u_2 (N-507 temperatura de topo)	127
Figura 24	Simulação: caso industrial (CDU) - entrada u_3 (N-507 stripping) . .	127
Figura 25	Simulação: caso industrial (CDU) - entrada u_5 (Bombeamento de diesel pesado)	128
Figura 26	Simulação: caso industrial (CDU) - entrada u_6 (Refluxo de diesel) .	128
Figura 27	Simulação: caso industrial (CDU) - entrada u_7 (Fluxo de querosene)	129
Figura 28	Simulação: caso industrial (CDU) - saída y_1 (N-507 massa de vapor por volume de cru)	129
Figura 29	Simulação: caso industrial (CDU) - saída y_2 (Fluxo de nafta pesada)	130

Figura 30	Simulação: caso industrial (CDU) - saída y_3 (N-506 refluxo)	130
Figura 31	Simulação: caso industrial (CDU) - saída y_4 (Temperatura de retirada de querosene)	131
Figura 32	Simulação: caso industrial (CDU) - saída y_5 (Diesel ASTM D-86 95%)	131
Figura 33	Simulação: caso industrial (CDU) - saída y_6 (Flash-point do diesel)	132
Figura 34	Simulação: caso industrial (CDU) - saída y_7 (Ponto final nafta leve ASTM D-86)	132
Figura 35	Simulação: caso industrial (CDU) - saída y_8 (N-507 refluxo)	133
Figura 36	Simulação: caso industrial (CDU) - saída y_9 (Fluxo de nafta leve)	133
Figura 37	Simulação: caso industrial (CDU) - saída y_{10} (L-506 Carga térmica)	134
Figura 38	Função objetivo: caso industrial (CDU)	134

Lista de Tabelas

Tabela 1	Valores mínimos, máximos e iniciais para as saídas - Sistema 3×2	105
Tabela 2	Valores mínimos, máximos e iniciais para as entradas - Sistema 3×2	105
Tabela 3	Tabela com novos valores mínimos e máximos para as saídas - Sistema 3×2	106
Tabela 4	Definição e valores mínimos, máximos e iniciais para as saídas - CDU	123
Tabela 5	Definição e valores mínimos e máximos para as entradas - CDU	123
Tabela 6	Valores iniciais e desejados para as entradas - CDU	125
Tabela 7	Novos valores de zona para as saídas, restrições para as entradas e alteração de alvo para as entradas - CDU	125

Lista de siglas

CDU	<i>Crude Distillation Unit.</i>
DC-IHMPC	<i>Distributed and Cooperative Infinite Horizon Model Predictive Control.</i>
DCMPC	<i>Distributed and Cooperative Model Predictive Control.</i>
DMC	<i>Dynamic Matrix Control.</i>
DMPC	<i>Distributed Model Predictive Control.</i>
FT	Funções de Transferência.
IFAC	<i>International Federation of Automatic Control.</i>
IHMPC	<i>Infinity Horizon Model Predictive Control.</i>
LMI	<i>Linear Matrix Inequalities.</i>
LQR	<i>Linear Quadratic Regulator.</i>
LTI	<i>Linear time-invariant.</i>
MIMO	<i>multiple input, multiple output.</i>
MIQP	<i>Mixed Integer Quadratic Programming.</i>
MPC	<i>Model Predictive Control.</i>
OPOM	<i>Output Prediction Oriented Model.</i>
PID	<i>Proportional–Integral–Derivative Controller.</i>
QP	<i>Quadratic Programming.</i>
RTO	<i>Real Time Optimization.</i>
SISO	<i>single input, single output.</i>
SMOC	<i>Shell Multivariable Optimizing Controller.</i>

Lista de símbolos

$\mathbf{0}$	Matriz nula de dimensão qualquer (p. 36).
$\mathbf{A}$	Matriz de transição dos estados (eq. (1)).
$\mathbf{A}_{ii}$	Matriz de transição dos estados para sistemas descentralizados e distribuídos (eq. (66)).
$\mathbf{A}_{ij}$	Matriz de transição dos estados do sub-sistema j para o sub-sistema i em sistemas distribuídos (eq. (73)).
$\mathbf{B}$	Matriz que relaciona as entradas do sistema ao estado (eq. (1)).
$\mathbf{B}^d$	Matriz que relaciona as ações de controle para sistemas com tempo morto às variáveis x^d (eq. (29)).
$\mathbf{B}^s$	Matriz que relaciona as ações de controle para sistemas com tempo morto às variáveis x^s (eq. (29)).
$\mathbf{B}_{ii}$	Matriz que relaciona as entradas ao estado para sistemas descentralizados e distribuídos (eq. (66)).
$\mathbf{B}_{ij}$	Matriz que relaciona as entradas do sub-sistema j para o sub-sistema i em sistemas distribuídos (eq. (73)).
$\mathbf{C}$	Matriz que relaciona os estados às saídas do sistema (eq. (2)).
c	Termo constante da função objetivo (eq. (203)).
$\mathbf{C}_f$	Matriz para o termo linear da função objetivo (eq. (203)).
$\mathbf{C}_i$	Matriz que relaciona os estados às saídas para sistema descentralizados e distribuídos (eq. (67)).
$\mathbf{D}^0$	Matriz que contém todos os valores $d_{i,j}^0$ (eq. (13)).
$\mathbf{D}_{ij}^0$	Matriz que contém os valores $d_{i,j}^0$ para o sub-sistema i em relação ao sub-sistema j (eq. (83)).

$\mathbf{D}_{\mathbf{m}ij}^0$	Composição das matrizes $\mathbf{D}_{ij}^0$ (eq. (108)).
$\mathbf{D}^d$	Matriz que contém todos os valores $d_{i,j,k}^d$ (eq. (14)).
$\mathbf{D}_i^d$	Matriz que contém os valores $d_{i,j,k}^d$ para o sub-sistema i (eq. (83)).
$\tilde{\mathbf{D}}_i^u$	Matriz auxiliar utilizada para restrições lineares de igualdade (eq. (161)).
$\mathbf{D}_{\mathbf{m}i}^u$	Composição das matrizes identidade para a segunda soma na função objetivo da seção 5 (eq. (164)).
$d_{i,j,k}^d$	k -ésima resposta referente aos polos estáveis das funções de transferência $G_{i,j}(s)$ (eq. (6)).
$d_{i,j}^0$	Ganho das funções de transferência $G_{i,j}(s)$ (eq. (6)).
$\mathbf{F}$	Matriz que contém a dinâmica dos polos estáveis (p. (12)).
$\mathbf{F}_{\mathbf{d}uij}$	Composição de matrizes $\mathbf{D}_i^d$ e $\mathbf{N}_{ij}$ (eq. (112)).
$\mathbf{F}_{\mathbf{f}ui}$	Composição de matrizes $\mathbf{F}_i$ (eq. (111)).
$\mathbf{F}_i$	Matriz que contém a dinâmica dos polos estáveis para o sub-sistema i (eq. (83)).
$\mathbf{F}_{\mathbf{u}ij}$	Relação entre as matrizes $\mathbf{F}_{\mathbf{f}ui}$ e $\mathbf{F}_{\mathbf{d}uij}$ (eq (113)).
$\mathbf{F}_{\mathbf{x}i}$	Composição de matrizes $\mathbf{F}_i$ (eq. (110)).
$\mathbf{G}(s)$	Funções de transferência que representam um sistema (eq. (3)).
$\mathbf{H}$	Na seção 2.5.1 matriz hessiana de uma função f qualquer (eq. (56)). Nas outras seções, matriz para o termo quadrático da função objetivo (eq. (203)).
$\bar{\mathbf{I}}$	Composição qualquer de matrizes identidade (eq. (119)).
$\bar{\mathbf{I}}_i$	Composição de matrizes identidade para o sub-sistema i (eq. (109)).
$\mathbf{I}_n$	Matriz identidade de dimensão n (p. 36).
$\mathbf{J}$	Matriz auxiliar para relacionar o estado e as entradas (eq. (16)).
k	Instante atual (p. 34).
$\mathbf{L}$	Matriz de ganho do observador (eq. (42)).

M	Número de sub-sistemas em um esquema de controle descentralizado ou distribuído (p. 53).
m	Horizonte de controle (eq. (49)).
$\mathbf{N}$	Matriz auxiliar para correspondência estado/entrada (eq. (15)).
$\mathbf{N}_{ij}$	Matriz auxiliar para correspondência estado i /entrada j (eq. (83)).
na	Ordem das funções de transferência $G_{i,j}(s)$ (p. 34).
nb	Ordem do numerador das funções de transferência $G_{i,j}(s)$ (p. 34).
nd	Dimensão da componente $d_{i,j,k}^d$; $nd = ny \times nu \times na$ (p. 36).
nu	Número de entradas (p. 34).
nu_i	Número de entradas para o sub-sistema i (p. 53).
nx	Dimensão do estado (p. 34).
nx_i	Dimensão do estado para o sub-sistema i (p. 53).
ny	Número de saídas (p. 34).
ny_i	Número de saídas para o sub-sistema i (p. 53).
$\mathbf{P}$	Matriz de covariância do erro de estimação para o filtro de Kalman (eq. (48)).
$\mathbf{Q}$	Matriz de pesos para a priorização das saídas (eq. (49)).
q	Horizonte de predição (eq. (49)).
$\mathbf{Q}_{gi}^u$	Composição das matrizes $\mathbf{Q}_{ui}$ (eq. (165)).
$\mathbf{Q}_{gi}$	Composição das matrizes $\mathbf{Q}_i$ (eq. (114)).
$\mathbf{Q}_i$	Matrizes de pesos para a priorização de saídas em um arranjo distribuído (eq. (97)).
$\mathbf{Q}_u$	Matriz de pesos para as entradas com RTO (eq. (64)).
$\mathbf{Q}_{ui}$	Matriz de pesos para as entradas com RTO em um arranjo distribuído (eq. (158)).

$\mathbf{Q}_y$	Matriz de pesos para as saídas com RTO (eq. (64)).
$\mathbf{Q}_{yi}$	Matriz de pesos para as saídas com RTO em um arranjo distribuído (eq. (158)).
$\mathbf{R}$	Matrizes de penalização para movimentos nas entradas (eq. (49)).
$\mathbf{R}_1$	Matriz de covariância no erro do modelo para o filtro de Kalman (p. 45).
$\mathbf{R}_2$	Matriz de covariância no erro das medidas para o filtro de Kalman (p. 45).
$\mathbf{R}_{gi}$	Composição das matrizes $\mathbf{R}_i$ (eq. (116)).
$\mathbf{R}_i$	Matrizes de penalização para movimentos nas entradas em um arranjo distribuído (eq. (97)).
$\mathbf{S}$	Matriz de peso para as variáveis de folga (eq. (54)).
s	Variável da transformada de Laplace (p. 34).
$\mathbf{S}_i$	Matrizes de peso para as variáveis de folga em um arranjo distribuído (eq. (97)).
$S_{i,j}(s)$	Resposta ao degrau da função de transferência (eq. (6)).
T_i	Período de amostragem para o sub-sistema i em sistemas com diferentes escalas de tempo (eq. (194)).
$\mathbf{U}$	Sequência de entradas em um sistema qualquer (eq. (38)).
$\mathbf{u}$	Vetor de entradas do sistema (p. 34).
$U(t)$	Função degrau (eq. (7)).
$\mathbf{u}^{des}$	Alvos (set-points) nas entradas para arranjo com RTO (p. 50).
$\mathbf{u}^{max}$	Restrições de valores máximos referentes às entradas (eq. (50)).
$\mathbf{u}_i^{max}$	Restrições de valores máximos referentes às entradas do sub-sistema i (p. 55).
$\mathbf{u}^{min}$	Restrições de valores mínimos referentes às entradas (eq. (50)).
$\mathbf{u}_i^{min}$	Restrições de valores mínimos referentes às entradas do sub-sistema i (p. 55).

$\mathcal{U}^{nu}$	Conjunto convexo de valores admissíveis para as entradas (p. 48).
$\mathcal{U}^{nu_i}$	Conjunto convexo de valores admissíveis para as entradas do sub-sistema i (p. 55).
$\mathcal{U}_{eq}^{nu_i}$	Conjunto convexo de valores admissíveis para as entradas do sub-sistema i definido por equações (p. 75).
$\mathcal{U}_{in}^{nu_i}$	Conjunto convexo de valores admissíveis para as entradas do sub-sistema i definido por restrições compostas por inequações (p. 75).
$\mathbf{u}_i$	Vetor de entradas para o sub-sistema i em sistemas descentralizados ou distribuídos (eq. (66)).
$\mathbf{u}_k$	Vetor de entradas do sistema para o horizonte de predição m (p. 55).
$\mathbf{u}_{ki}$	Vetor de entradas do sub-sistema i para o horizonte de predição m (eq. (78)).
$\mathbf{v}(k)$	Ruído aleatório nas medidas das saídas utilizado no filtro de Kalman (eq. (45)).
V_k	Função custo para todo o sistema calculada no instante k (eq. (49)).
V_{ki}	Função custo para o sub-sistema i calculada no instante k no caso não cooperativo (eq. (77a)).
$\mathbf{w}(k)$	Ruído aleatório no modelo do estado utilizado no filtro de Kalman (eq. (44)).
$\mathbf{W}_c$	Matriz de controlabilidade (eq. (37)).
w_i	Pesos para combinação convexa das funções custo de cada sub-sistema (eq. (82b)).
$\mathbf{W}_o$	Matriz de observabilidade (eq. (40)).
$\mathbf{x}$	Vetor de estados do sistema (p. 34).
$\mathbf{x}^d$	Estado que prevê a evolução dinâmica do sistema (eq. (10)).
$\mathbf{x}_i^d$	Estado que prevê a evolução dinâmica do sub-sistema i (eq. (83)).
$\mathbf{x}^s$	Predição da saída do sistema em estado estacionário (eq. (10)).

$\mathbf{x}_i^s$	Predição da saída do sub-sistema i em estado estacionário (eq. (83)).
$\mathbf{x}_i$	Vetor de estados para o sub-sistema i em sistemas descentralizados ou distribuídos (eq. (66)).
$\mathbf{y}$	Vetor de saídas do sistema (p. 34).
$\mathbf{y}^{max}$	Valores máximos aceitáveis para as saídas na operação por faixas (p. 50).
$\mathbf{y}^{min}$	Valores mínimos aceitáveis para as saídas na operação por faixas (p. 50).
$\mathbf{y}_k^{sp}$	Set-point para as saídas (eq. (49)).
$\mathbf{y}_{ki}^{sp}$	Set-point para as saídas do sub-sistema i (eq. (97)).
$\mathbf{y}_i$	Vetor de saídas para o sub-sistema i em sistemas distribuídos (eq. (67)).

Lista de símbolos - alfabeto grego

α	Na sessão 2.2.2: parte real do resíduo de uma função de transferência com polos complexos conjugados (eq. (20)). Na sessão 7.2, função “filtro de instantes de amostragem” (eq. (195)).
β	Parte imaginária do resíduo de uma função de transferência com polos complexos conjugados (eq. (20)).
$\bar{\Delta}\mathbf{u}_{kj}$	Diferenças entre $\Delta\mathbf{u}_{kj}^\infty$ e $\Delta\mathbf{u}_{kj}^*$, $j = 1, \dots, M$ (p. 80).
δ	Variável de folga (em inglês: “slacks”) para sistemas centralizados (eq. (54)).
δ_{kii}	Variável de folga (em inglês: “slacks”) para o sub-sistema i em sistemas descentralizados (eq. (97)).
$\bar{\delta}_{ki}$	Vetor de variáveis de folga para cada sub-sistema i (eq. (106)).
Δt	Intervalo de amostragem (eq. (6)).
$\tilde{\delta}_{k+1i}$	Vetor de variáveis de folga para cada sub-sistema i deslocado para o instante $k + 1$ (p. 86).
$\bar{\delta}_{kj}$	Vetor de variáveis de folga para cada sub-sistema j , $j \neq i$ (Algoritmo 1 passo 4.a) .
$\Delta\mathbf{u}$	Vetor de ação de controle: variação na entrada (eq. (8)).
$\Delta\mathbf{u}^{max}$	Restrições de maior ação de controle possível para as entradas (eq. (51)).
$\Delta\mathbf{u}_i^{max}$	Restrições de maior ação de controle possível para as entradas do sub-sistema i em sistemas descentralizados e distribuídos (eq. (121)).
$\Delta\mathbf{u}_i$	Vetor de ação de controle: variação nas entradas do sub-sistemas i em sistemas descentralizados e distribuídos (eq. (83)).
$\Delta\mathbf{u}_{ki}$	Vetor de ação de controle: variação nas entradas do sub-sistemas i ao longo do horizonte de predição m (eq. (107)).

$\Delta \tilde{\mathbf{u}}_{k+1i}$	Vetor de ação de controle: variação nas entradas do sub-sistemas i deslocado para $k + 1$ (p. 86).
$\Delta \mathbf{u}_{kj}$	Vetor de ação de controle: variação nas entradas do sub-sistemas j ao longo do horizonte de predição m (eq. (118)).
$\Delta \mathbf{v}_{kj}$	Vetores de ação de controle para $j = 1, \dots, M$: variações nas entradas para valores de p muito grandes (eq. (135)).
$\Delta \mathbf{z}_{kj}$	Vetores de ação de controle para $j = 1, \dots, M$: variações nas entradas para valores de p muito grandes (eq. (135)).
$\bar{\Delta} \bar{\delta}_{kj}$	Diferenças entre $\bar{\delta}_{kj}^\infty$ e $\bar{\delta}_{kj}^*$, $j = 1, \dots, M$ (p. 80).
$\bar{\kappa}_{kj}$	Valores de variável de folga para p muito grande (eq. (135)).
$\bar{\lambda}_{kj}$	Valores de variável de folga para p muito grande (eq. (135)).
λ	Números reais usados para definição de convexidade (p. 48).
ω	Parte imaginária dos polos de uma função de transferência com polos complexos conjugados (eq. (20)).
Φ	Matriz auxiliar na construção de Ψ (eq. (18)).
Ψ	Matriz que relaciona as saídas aos polos estáveis (eq. (17)).
Ψ_i	Matriz que relaciona as saídas aos polos estáveis para o sub-sistema i (eq. (84)).
$\Psi_{\mathbf{g}i}$	Composição das matrizes Ψ_i (eq. (117)).
$\theta_{\max}$	Maior tempo morto entre as entradas e saídas (eq. (29)).
$\theta_{\max}^i$	Maior tempo morto entre as entradas e saídas para o sub-sistema i (p. 109).

Sumário

1	Introdução	27
2	Revisão Bibliográfica	30
2.1	Histórico dos controladores preditivos	30
2.2	Modelos em espaço de estados	33
2.2.1	<i>Output Prediction Oriented Model</i> (OPOM)	34
2.2.2	Modelo contendo espaço de estados estritamente reais	38
2.2.3	OPOM para sistemas com tempo morto	40
2.3	Controlabilidade, alcançabilidade, observabilidade e detectabilidade	41
2.4	Observadores de estado e o filtro de Kalman	43
2.4.1	Obtenção do estado através de estimadores	44
2.4.2	O filtro de Kalman	45
2.5	O controlador preditivo baseado em modelo (MPC)	46
2.5.1	Convexidade das funções objetivo e dos conjuntos gerados pelas restrições	48
2.6	O MPC em uma estrutura de camadas de otimização	50
2.6.1	O MPC para sistemas com tempo morto	52
2.7	O controlador preditivo baseado em modelo distribuído (DMPC)	53
2.7.1	Diferenças entre os modelos de estado para diferentes tipos de sistema	53
2.7.2	Diferenças entre controladores para cada grau de centralização	55
3	Modelos em espaço de estados para sistemas distribuídos	59
3.1	OPOM adaptado para sistemas distribuídos	59
3.2	OPOM adaptado para sistemas distribuídos com tempo morto	62
3.3	Controlabilidade, alcançabilidade, observabilidade e detectabilidade dos modelos distribuídos	64
4	Proposta para o controlador distribuído: DC-IHMPC	66
4.1	Formulação da função objetivo e das restrições para o DC-IHMPC	66
4.1.1	Cálculo iterativo para obtenção de solução ótima	72
4.1.2	Esquema de cascadeamento para solução ótima	74
4.2	Factibilidade recursiva e estabilidade do problema proposto	75
4.2.1	Convexidade do conjunto $\mathcal{U}^{nu}$ e das funções objetivo V_k	75
4.2.2	Convergência do modelo iterativo	76
4.2.3	Convergência do modelo iterativo no arranjo cascadeado	81
4.2.4	A questão das restrições e o modelo iterativo	82

4.2.5	Factibilidade recursiva do controlador proposto	85
4.2.6	Estabilidade do controlador proposto	86
4.3	Resultados de simulação para sistemas simplificados	87
5	Proposta para o controlador DC-IHMPC com RTO	94
5.1	Formulação da função objetivo e das restrições para o DC-IHMPC com RTO	94
5.1.1	Cálculo iterativo para obtenção de solução ótima	98
5.2	Factibilidade recursiva e estabilidade do problema proposto com RTO . . .	99
5.2.1	A questão das restrições e o modelo iterativo com RTO	100
5.2.2	Factibilidade recursiva do controlador proposto com RTO	102
5.2.3	Estabilidade do controlador proposto com RTO	103
5.3	Resultados de simulação para sistemas simplificados com RTO	104
6	Proposta para o controlador DC-IHMPC com tempo morto	109
6.1	Formulação da função objetivo e das restrições para o DC-IHMPC com tempo morto	109
6.2	Resultados de simulação para sistemas simplificados com tempo morto	111
7	Proposta para o controlador DC-IHMPC com diferentes escalas de tempo	115
7.1	As diferentes escalas de tempo	116
7.2	Formulação da função objetivo e das restrições para o DC-IHMPC com múltiplas escalas de tempo	117
7.3	Resultados de simulação para sistemas simplificados com múltiplas escalas temporais	118
8	Resultados de simulação para um caso industrial	122
9	Considerações finais	136
9.1	Conclusões	136
9.2	Sugestões para trabalhos futuros	137
Apêndice A - Função objetivo para o controlador definido no Capítulo 4 em sua forma matricial		143
Apêndice B - Função objetivo para o controlador definido no Capítulo 5 em sua forma matricial		146
Apêndice C - Funções de transferência (CDU)		150

1 Introdução

Para garantir uma operação segura e rentável de uma planta química, bem como atender a pré-requisitos de qualidade de produto e legislação ambiental, são necessárias estruturas de controle de processos. Em plantas complexas, estruturas clássicas de controle regulatório podem ser insuficientes para garantir o controle, ou então para garantir eficiência econômica. De acordo com Rawlings e Stewart [1], a complexidade das plantas químicas aumentou significativamente nas últimas décadas, visando garantir eficiência econômica através de projetos mais complexos e operações interligadas. Deste modo, uma planta que poderia ser operada na década de 1950 manualmente por operadores seria impraticável na conjuntura atual.

As primeiras tecnologias que surgiram para se somar às estratégias clássicas de controle nestas situações foram o “*Dynamic Matrix Control* (DMC)” e posteriormente o controle preditivo baseado em modelo (em inglês: *Model Predictive Control* (MPC)). Tais tecnologias vêm sendo desenvolvidas desde o seu surgimento até o estado da arte em que se encontram atualmente. Uma estrutura comum consiste em dividir o controlador em camadas, na qual a superior leva em conta modelos rigorosos e planejamento em estado estacionário para maximizar a lucratividade. A camada intermediária faz um controle supervisório, escolhendo as melhores ações de controle a serem tomadas a cada instante. Finalmente, a camada inferior faz o controle regulatório, mantendo a planta o mais próxima possível da condição ótima.

Fundamentalmente, o controle preditivo se baseia em um modelo da planta, normalmente obtido a partir de Funções de Transferência (FT) para estimar o estado futuro a partir de alterações nas variáveis manipuladas. Define-se uma função objetivo que leva em conta, basicamente, duas parcelas. A primeira delas é a distância entre o estado atual e o estado desejado, a segunda parcela é o esforço de controle necessário para se atingir tal situação. A solução deste problema é a de minimizar a função objetivo. Somente a primeira ação de controle calculada é implementada num determinado instante, de modo que todo o cálculo é repetido no instante seguinte. A isso dá-se o nome de horizonte deslizante (em inglês: *moving horizon*).

As principais vantagens do controle preditivo baseado em modelo são: possibilidade de lidar com restrições de processo, de incorporar tempos mortos ao modelo e de lidar com sistemas com várias entradas e várias saídas (em inglês: *multiple input, multiple output* - MIMO) de maneira muito similar àquela que se lida com sistemas de uma entrada e uma saída (em inglês: *single input, single output* - SISO). Há, entretanto, desvantagens

também em utilizar este tipo de estratégia. As principais são a necessidade de obtenção de um modelo confiável da planta, o que pode resultar em um esforço de alto custo e o problema de realizar otimização online para sistemas muito grandes, o que é custoso em termos de esforço computacional e instalação de infraestrutura (fibra óptica etc.).

Em Farina e Scatollini [2] são apresentados sistemas grandes, que funcionam em larga escala, dificultando a estratégia de controle preditivo baseado em modelo. Entre eles estão redes de transmissão de eletricidade, redes de transporte e hidroelétricas. A solução comumente empregada nestes casos é descentralizar o MPC em vários subsistemas. Em casos nos quais estes subsistemas interagem, o conjunto pode se tornar instável, o que levou a uma solução intermediária conhecida como MPC Distribuído, no qual cada controlador conhece a atuação de seus vizinhos (diferentemente do sistema descentralizado). Em Farina e Scatollini [2] é possível observar uma lista de propostas de controladores deste estilo com diversas características distintas. Há, por exemplo, aqueles considerados “cooperativos”, o que significa que eles não somente conhecem a atuação de seus vizinhos, mas cooperam para a obtenção de uma solução.

O presente trabalho teve por objetivo desenvolver diversos controladores de horizonte infinito, todos distribuídos e cooperativos. Eles foram genericamente intitulados “*Distributed and Cooperative Infinite Horizon Model Predictive Control* (DC-IHMPC)”. Tais controladores têm todos estabilidade garantida e utilizam incrementos nas entradas. Além disso, eles apresentam variáveis de folga (até agora pouco exploradas na literatura para sistemas distribuídos) para garantir a existência de solução a cada instante. Mais ainda, tais variáveis foram adicionadas de forma cruzada entre diferentes subsistemas, o que garante que não seja necessário nenhum tipo de sintonia específica nos parâmetros para garantia de estabilidade (como é o caso de Santana *et al.* [3]). Finalmente, cada controlador lida com uma situação em específico: o primeiro deles atua com “set-points” nas saídas em sistemas quadrados; o segundo atua junto a outras camadas de otimização (RTO), faixas nas saídas e alvos nas entradas e sistemas não quadrados; o terceiro é uma adaptação do segundo para lidar com tempo morto; finalmente, o quarto é um controlador que lida com múltiplas escalas temporais. Tais controladores foram pouco ou nada abordados na literatura até o presente momento.

Talvez, a principal vantagem destes controladores é poder dividir o controle tal que o mesmo atue através de vários servidores. Isso reduz o tempo de computação para calcular a resposta do atuador, de modo que o controlador pode, no futuro, embarcar várias tecnologias no mesmo software como inteligência artificial, controle de falhas etc. Tal possibilidade é mais remota em sistemas grandes e centralizados.

Este trabalho foi dividido em nove Capítulos. O Capítulo 2 tratou de uma revisão bi-

bliográfica acerca de modelos em espaço de estados e do tema de controle preditivo em geral. O Capítulo 3 apresentou os modelos de estado desenvolvidos para o controlador distribuído e cooperativo. Neste Capítulo, também foram abordadas as questões de controlabilidade, alcançabilidade, observabilidade e detectabilidade dos modelos distribuídos. No Capítulo 4, propôs-se e definiu-se o primeiro controlador citado acima. Neste Capítulo abordaram-se também as questões do modelo iterativo para a convergência de sistemas distribuídos, da convexidade, da factibilidade recursiva e da convergência do problema. Nos Capítulos seguintes (5 e 6) foram apresentados o segundo e o terceiro controladores distribuídos citados acima (acoplado a camadas de otimização e com tempo morto). Nestes Capítulos também foram discutidas questões de factibilidade recursiva e convergência. No Capítulo 7 abordou-se o último controlador (múltiplas escalas de tempo). No Capítulo 8 apresentou-se uma simulação de um caso industrial, a saber, uma planta de destilação de petróleo cru, mostrando o desempenho dos sistemas aqui desenvolvidos em uma situação da indústria. Finalmente, no Capítulo 9 foram feitas as considerações finais: uma conclusão e sugestões de continuidade do trabalho. Todas as simulações apresentadas neste trabalho foram feitas em MATLABTM R2015a e Intel[®]CoreTMi5-7200U CPU@2.50GHz

Quanto à nomenclatura, uma notação padrão foi utilizada: letras maiúsculas em negrito representam matrizes. Letras minúsculas em negrito representam vetores-coluna. Para uma matriz quadrada, $\mathbf{Q}$, $\mathbf{Q} \geq \mathbf{0}$ representa uma matriz positiva semi-definida e $\mathbf{Q} > \mathbf{0}$ representa uma matriz positiva definida. $\|\mathbf{x}\|_{\mathbf{Q}}^2$ representa $\mathbf{x}^T \mathbf{Q} \mathbf{x}$. Além disso, para qualquer variável $\mathbf{x}$ variante no tempo, $\mathbf{x}(k+l|k)$ representa a predição desta variável para o instante $k+l$ baseada nas condições do instante k .

2 Revisão Bibliográfica

Dada a motivação do trabalho no capítulo anterior, pôde-se neste capítulo apresentar uma revisão bibliográfica. Esta foi escrita com o objetivo de contextualizar a tecnologia do controle preditivo baseado em modelo e esclarecer as bases para o presente trabalho. Para isso, o primeiro passo (seção 2.1) foi contextualizar o processo de desenvolvimento do MPC, as contribuições do grupo de pesquisa do professor Darci Odloak e alguma bibliografia acerca de controladores distribuídos permitindo familiarizar o leitor com esta tecnologia e abrir espaço para os desenvolvimentos deste trabalho.

Na sequência, mencionaram-se mais detalhadamente os modelos em espaço de estados (seção 2.2), em particular o OPOM (subseção 2.2.1), questões sobre controlabilidade (seção 2.3) e os estimadores de estado (seção 2.4).

A seguir, definiu-se detalhadamente o MPC centralizado (seção 2.5). Nesta seção, abordou-se também a questão de convexidade de conjuntos e funções (subseção 2.5.1). O MPC que atua junto com RTO foi apresentado na seção seguinte (2.6).

Finalmente, os controladores distribuídos foram abordados (seção 2.7). Nesta seção mencionaram-se os diferentes graus de centralização de modelos de estado (subseção 2.7.1) e abordaram-se as diferenças de funções objetivo para controladores centralizados e distribuídos (subseção 2.7.2) de modo a se ter uma base sobre a qual construir o MPC distribuído desenvolvido neste trabalho.

2.1 Histórico dos controladores preditivos

A primeira referência que se pode citar acerca de um controlador preditivo para múltiplas entradas e saídas é atribuída a Richelet *et al.* em simpósio da *International Federation of Automatic Control* (IFAC) (1976) [4] e depois detalhada em um artigo publicado pela Automatica [5]. É curioso notar que já nesta época, os autores pensavam em uma estrutura em camadas, imaginando controladores do tipo proporcional, integral e derivativo (PID) controlando diretamente válvulas e camadas superiores realizando otimização de processos. De fato, quase simultaneamente foi desenvolvida uma estratégia para otimi-

zação. Esta passou a ser utilizada para controle e é a chamada *Dynamic Matrix Control* (DMC), desenvolvida pela Shell [6–8]. Sua abordagem é a de um problema de mínimos quadrados baseado em um modelo de resposta a um degrau/impulso e a diferença entre a trajetória prevista e um set-point.

Posteriormente, adaptações transformando o problema do DMC foram apresentadas [9, 10]. A seguir, engenheiros da Shell desenvolveram outra abordagem que controla sistemas até mesmo com polos instáveis (desde que um observador de estados adequado seja usado) e com tempo morto, o chamado *Shell Multivariable Optimizing Controller* (SMOC) [11]. Uma alternativa ao uso de resposta ao impulso é a utilização de espaço de estados. Tal abordagem passou a ser padrão na literatura de controle preditivo [12]. Tais sistemas, além de economizarem informação (as matrizes do espaço de estados são bem mais compactas que a matriz do DMC), permitem o uso dos conhecimentos clássicos da teoria de controle linear.

Adicionalmente, Michalska e Mayne [13] especificam restrições sobre as condições do estado final, definindo um conjunto $\mathcal{S}$ (invariante) dentro do qual o estado final deve se encontrar, de modo que o mesmo permaneça dentro de $\mathcal{S}$ sempre. Este tipo de MPC é conhecido como “Dual”.

Nos anos 1990, a Honeywell apresentou uma tecnologia que mantém o controle por faixas aceitáveis ao invés de um set-point [14]. Isso é interessante para algumas variáveis, por exemplo, nível de tanque, que não precisam ser mantidas em um valor específico, mas sim em uma faixa aceitável. Rawlings e Muske [15, 16] foram os primeiros a propor um controlador com horizonte de predição infinito (IH MPC) e demonstraram que, com isso, garante-se estabilidade.

Como sempre há incertezas no modelo, não é possível determinar exatamente o valor de uma saída baseado em um valor de entrada. Lee, Morari e Garcia [17] contornaram inicialmente este problema projetando um modelo de estado incremental. Tal situação gera entretanto modos integradores, mesmo em sistemas que são estáveis em malha aberta. Finalmente, Badgwell [18] apresenta em 1997 uma forma inovadora de lidar com incertezas no modelo do estado que leva ao desenvolvimento dos chamados sistemas robustos.

Passando para o trabalho do grupo de pesquisa do prof. Darci Odloak, o mesmo tem desenvolvido seu trabalho em diversas das frentes citadas acima. Em destaque está o fato de os sistemas desenvolvidos serem normalmente aptos para utilização em processos industriais. Um bom exemplo disso é encontrado em Martin, Zanin e Odloak [19], no qual uma coluna de destilação de petróleo cru é controlada com um MPC de horizonte infinito.

Dentre as suas principais contribuições, está um modelo de estado chamado *Output Pre-*

dition Oriented Model (OPOM) [20], que permite facilmente o cálculo de um estado terminal para MPC de horizonte infinito. Dentre os controladores mais importantes desenvolvidos, estão aqueles com tempo morto (sistemas nos quais decorre um período de tempo entre a ação de controle e a resposta do sistema - são comuns na engenharia química por conta do grande porte dos equipamentos), desenvolvido por Santoro e Odloak [21]. Há também sistemas com polos instáveis e processos integradores, por Sotomayor e Odloak [22], e também por Martins e Odloak [23] e Carrapiço e Odloak [24]. Neste último, foram propostas duas estratégias: zerar os polos integradores usando folgas que devem tender a zero, ou dividir o problema em dois, sendo que num primeiro momento zeram-se os polos integradores e no segundo momento o resto da função objetivo.

Além disso, foram desenvolvidos pelo grupo controladores que consideram faixas de operação e integração com o chamado “*Real Time Optimization* (RTO)” - sistemas rigorosos não lineares calculados a partir de estados estacionários e ótimos do ponto de vista econômico, o que permite trabalhar com sistemas não quadrados, por González e Odloak [25].

O grupo também desenvolve adaptações para sistemas robustos, ou seja, aqueles que visam levar em conta incertezas no modelo. Um exemplo disso são sistemas incertos com tempo morto, por González e Odloak [26]. Outros exemplos são a integração com RTO, por González, Marchetti e Odloak [27] e também por Alvarez e Odloak [28] (o primeiro trabalho utiliza um esquema de duas camadas de controle, enquanto que o segundo trabalho utiliza três camadas). Dual MPCs robustos também foram desenvolvidos por Capron e Odloak utilizando o chamado “*Linear Quadratic Regulator* (LQR)” e as “*Linear Matrix Inequalities* (LMI)” [29].

Em resumo, estes são os principais temas abordados pelo grupo de pesquisa. Um tema ainda pouco abordado pelo grupo é o MPC distribuído ou “*Distributed Model Predictive Control* (DMPC)”, objetivo deste mestrado. Pensados para funcionar em sistemas de grande porte, os DMPCs têm por objetivo atuarem como intermediários entre um sistema completamente descentralizado e um sistema centralizado. O sistema descentralizado pode resultar instável caso as interações entre as diversas partes dos sistemas sejam fortes. O sistema centralizado tem a desvantagem de que, em sistemas muito grandes, há aumento significativo de custo para montar o aparato de comunicação de cada uma de suas partes, pode haver uma perda total do sistema de controle caso haja uma falha num servidor ou outro equipamento ou então o tempo de computação fica muito grande, especialmente para sistemas não lineares. Alguns destes problemas e propostas de controladores distribuídos são apresentadas por Stewart *et al.* [30] e Venkat *et al.* [31].

Como foi dito, os sistemas podem ser classificados em centralizados, distribuídos e descentralizados. Via de regra, os sistemas distribuídos podem ser subclassificados em não-

cooperativos e cooperativos, sendo os segundos denominados “*Distributed and Cooperative Model Predictive Control* (DCMPC)”. Os primeiros podem, em determinados casos, gerar instabilidade por atingirem um ponto ótimo de Nash. Os segundos dificilmente mostram-se instáveis por convergirem para um ponto ótimo de Pareto. Scattolini [32] fez uma revisão acerca dos tipos de controlador e dos arranjos em camadas para integração, por exemplo, com RTO.

Sistemas não-cooperativos nem sempre são inadequados. Em alguns casos, a interação entre eles gera um sistema controlável, ou então o ótimo de Nash é adequado. Farina *et al.* [2, 33] desenvolveram um sistema não cooperativo no qual o fato de eles não cooperarem simplifica as conexões entre os mesmos.

Quanto a modelos cooperativos, vários algoritmos foram desenvolvidos por diversos grupos de pesquisa. Rawlings e Stewart [1], por exemplo, dão heurísticas para pareamento de subsistemas. Trodden, Monasterios e Maestre [34] tentam reduzir distúrbios mútuos. Mendes *et al.* [35] utilizam um conjunto de variáveis binárias e contínuas em um esquema de programação mista: *Mixed Integer Quadratic Programming* (MIQP). Liu *et al.* [36] trabalham com modelos não lineares. Al-Gherwi, Budman e Elkamel [37] mostram uma primeira abordagem de um sistema distribuído robusto, no qual eles consideram dois sistemas com ganhos diferentes.

O presente trabalho desenvolveu um controlador preditivo baseado em modelo de horizonte infinito distribuído e cooperativo que pode operar considerando-se o caso tradicional de set-points nas saídas; o caso de sistemas não quadrados com faixas nas saídas; o caso de integração com RTO; o caso de sistemas com tempo morto e finalmente, o caso de sistemas com diferentes escalas de tempo. Antes de abordar efetivamente os controladores distribuídos, cabe falar dos modelos em espaço de estados e dos controladores centralizados.

2.2 Modelos em espaço de estados

Um sistema dinâmico pode ser representado por um modelo matemático, que por sua vez, deve representar o sistema fielmente. Tais modelos apresentam-se muitas vezes na forma de equações diferenciais [38]. Muitas vezes estas equações podem ser escritas na forma de um modelo de estado. O estado de um sistema dinâmico é o menor conjunto de variáveis cujo conhecimento em $t = t_0$ determina o comportamento do sistema em

$t \geq t_0$ na ausência de distúrbios ou outras influências [39]. O vetor de variáveis de estado é genericamente chamado de $\mathbf{x}(t)$ e tem dimensão nx . O estado pode ser influenciado por grandezas genericamente chamadas de “entradas” e denominadas por $\mathbf{u}(t)$. Além disso, podem ser definidas variáveis chamadas de saídas, função do estado e das entradas. Tais variáveis são genericamente chamadas de $\mathbf{y}(t)$. O espaço de dimensão nx cujos eixos coordenados são as variáveis de estado é chamado de espaço de estados [39].

As equações diferenciais que modelam um sistema em espaço de estados cujos coeficientes são constantes, ou dependentes somente da variável independente, são chamadas de lineares. Quando os coeficientes são constantes, diz-se que a equação diferencial é também invariante no tempo, ou então, utiliza-se a sigla em inglês: *Linear time-invariant* (LTI). Sistemas dinâmicos compostos por componentes LTI podem ser representados por tais equações, e são chamados de “Sistemas LTI”. Tais sistemas também podem ser representados por funções de transferência e por sua vez, por matrizes e vetores que compõem o estado do sistema [38]. Seja uma modelagem discreta no tempo e k o instante atual, então:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (1)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (2)$$

2.2.1 *Output Prediction Oriented Model* (OPOM)

Há diversas formas de se obter as matrizes $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ [39]. Particularmente, Odloak [20] propõe uma forma baseada em um degrau nas entradas, o chamado *Output Prediction Oriented Model* (OPOM). Considere um sistema com múltiplas entradas e múltiplas saídas ou, em inglês, “*multiple input, multiple output* (MIMO)” qualquer, sem polos integradores, inicialmente sem tempo morto e com nu variáveis de entrada e ny variáveis de saída. Tal sistema pode ser representado por um modelo de funções de transferência $\mathbf{G}(s)$, no qual s é uma variável no domínio da frequência:

$$\mathbf{G}(s) = \begin{bmatrix} G_{1,1}(s) & \cdots & G_{1,nu}(s) \\ \vdots & \ddots & \vdots \\ G_{ny,1}(s) & \cdots & G_{ny,nu}(s) \end{bmatrix} \quad (3)$$

Seja $G_{i,j}(s)$ a função de transferência que relaciona uma determinada saída $\mathbf{y}_i$ e uma determinada entrada $\mathbf{u}_j$. Sejam também na o grau do denominador e nb a ordem do numerador de todas as funções de transferência. Dessa forma, $G_{i,j}(s)$ pode ser escrita como:

$$G_{i,j}(s) = \frac{b_{i,j,0} + b_{i,j,1}s + b_{i,j,2}s^2 + \dots + b_{i,j,nb}s^{nb}}{(s - r_{i,j,1})(s - r_{i,j,2}) \dots (s - r_{i,j,na})} \quad (4)$$

Quando sujeita a um degrau unitário, a resposta desta função de transferência pode ser escrita como:

$$S_{i,j}(s) = \frac{G_{i,j}(s)}{s} = \frac{d_{i,j}^0}{s} + \frac{d_{i,j,1}^d}{s - r_{i,j,1}} + \dots + \frac{d_{i,j,na}^d}{s - r_{i,j,na}} \quad (5)$$

Sejam Δt o período de amostragem, $d_{i,j}^0$ o coeficiente estacionário e $d_{i,j,k}^d$ o k -ésimo coeficiente da resposta dinâmica referente aos polos estáveis, então (5) pode ser reescrita da seguinte forma:

$$S_{i,j}(k\Delta t) = (d_{i,j}^0 + d_{i,j,1}^d e^{r_{i,j,1}k\Delta t} + \dots + d_{i,j,na}^d e^{r_{i,j,na}k\Delta t}) U(t) \quad (6)$$

Na qual $U(t)$ é a função degrau unitário tal que:

$$U(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (7)$$

Considerando os valores da resposta ao degrau da função, obtêm-se exatamente os coeficientes que se enquadram na denominação acima de LTI, de modo que se pode escrever um estado do sistema na forma das equações (1) e (2) através destes coeficientes. A única diferença, entretanto, é que como se utilizou um degrau nas entradas, a forma de se escrever as equações para o estado difere de (1) por utilizar variação nas entradas, e não o próprio valor das mesmas. Isto, como já comentado na seção 2.1, tem a desvantagem de gerar polos integradores mesmo em sistemas com estabilidade em malha aberta, o que pode ser contornado e gera a grande vantagem de não ligar intrinsecamente o valor de uma entrada a uma variável de estado específica, dado que sempre pode haver incertezas nos modelos. As equações abaixo ilustram o modelo gerado pelo OPOM. Para mais detalhes acerca de sua construção, vide Odloak [20].

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \quad (8)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (9)$$

O vetor $\mathbf{x}$ pode ser dividido em duas partes $\mathbf{x} = [\mathbf{x}^s \ \mathbf{x}^d]^T$, sendo que $\mathbf{x}^s$ corresponde à predição da saída do sistema em estado estacionário (é relacionado com os modos integradores gerados pela forma incremental do estado) e tem dimensões $\mathbf{x}^s \in \mathbb{R}^{ny}$ e $\mathbf{x}^d$ corresponde aos modos estáveis do sistema original e tem dimensões $\mathbf{x}^d \in \mathbb{C}^{nd}$, de modo que $nd = ny \cdot nu \cdot na$. $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ podem ser reescritas também da seguinte forma:

$$\begin{bmatrix} \mathbf{x}^s(k+1) \\ \mathbf{x}^d(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{ny} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{x}^s(k) \\ \mathbf{x}^d(k) \end{bmatrix} + \begin{bmatrix} \mathbf{D}^0 \\ \mathbf{D}^d \mathbf{F} \mathbf{N} \end{bmatrix} \Delta\mathbf{u}(k) \quad (10)$$

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{I}_{ny} & \mathbf{\Psi} \end{bmatrix} \begin{bmatrix} \mathbf{x}^s(k) \\ \mathbf{x}^d(k) \end{bmatrix} \quad (11)$$

Em que $\mathbf{0}$ representa uma matriz nula de dimensões adequadas (por abuso de notação, assume-se que sempre que esta matriz aparece, ela tem as dimensões adequadas) e $\mathbf{I}_n$ representa a matriz identidade de dimensão n . $\mathbf{F}$, $\mathbf{D}^0$ e $\mathbf{D}^d$ são conforme se segue:

$$\mathbf{F} = \text{diag} \left(e^{r_{1,1,1}\Delta t} \dots e^{r_{1,1,na}\Delta t} \dots e^{r_{1,nu,1}\Delta t} \dots e^{r_{1,nu,na}\Delta t} \dots \right. \\ \left. e^{r_{ny,1,1}\Delta t} \dots e^{r_{ny,1,na}\Delta t} \dots e^{r_{ny,nu,1}\Delta t} \dots e^{r_{ny,nu,na}\Delta t} \right) \quad (12)$$

$$\mathbf{D}^0 = \begin{bmatrix} d_{1,1}^0 & \dots & d_{1,nu}^0 \\ \vdots & \ddots & \vdots \\ d_{ny,1}^0 & \dots & d_{ny,nu}^0 \end{bmatrix} \quad (13)$$

$$\mathbf{D}^d = \text{diag} \left(d_{1,1,1}^d \dots d_{1,1,na}^d \dots d_{1,nu,1}^d \dots \right. \\ \left. d_{1,nu,na}^d \dots d_{ny,1,1}^d \dots d_{ny,1,na}^d \dots d_{ny,nu,1}^d \dots d_{ny,nu,na}^d \right) \quad (14)$$

Repare que a matriz $\mathbf{A}$ no OPOM é diagonal. As dimensões destas matrizes são tais que: $\mathbf{D}^0 \in \mathbb{R}^{ny \times nu}$, $\mathbf{F} \in \mathbb{C}^{nd \times nd}$, $\mathbf{D}^d \in \mathbb{C}^{nd \times nd}$. Já $\mathbf{\Psi}$ e $\mathbf{N}$ são dadas por:

$$\mathbf{N} = \begin{bmatrix} \mathbf{J} \\ \mathbf{J} \\ \vdots \\ \mathbf{J} \end{bmatrix} \quad (15)$$

$$\mathbf{J} = \begin{bmatrix} \begin{matrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{matrix} \\ \vdots \\ \begin{matrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{matrix} \end{bmatrix} \quad (16)$$

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{\Phi} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{\Phi} \end{bmatrix} \quad (17)$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \quad (18)$$

As dimensões destas matrizes, por sua vez, são tais que: $\mathbf{J} \in \mathbb{R}^{nu \cdot na \times nu}$, $\mathbf{N} \in \mathbb{R}^{nd \times nu}$, $\mathbf{\Phi} \in \mathbb{R}^{nu \cdot na}$ e $\mathbf{\Psi} \in \mathbb{R}^{ny \times nd}$. Finalmente, o conjunto $\mathbf{D}^d \mathbf{FN} \in \mathbb{C}^{nd \times nu}$ pode ser escrito como:

$$\mathbf{D}^d \mathbf{FN} = \begin{bmatrix} d_{1,1,1}^d e^{r_{1,1,1} \Delta t} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ d_{1,1,na}^d e^{r_{1,1,na} \Delta t} & 0 & \cdots & 0 \\ 0 & d_{1,2,1}^d e^{r_{1,2,1} \Delta t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & d_{1,2,na}^d e^{r_{1,2,na} \Delta t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{ny,nu,1}^d e^{r_{ny,nu,1} \Delta t} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{ny,nu,na}^d e^{r_{ny,nu,na} \Delta t} \end{bmatrix} \quad (19)$$

2.2.2 Modelo contendo espaço de estados estritamente reais

A formulação apresentada na subseção 2.2.1 contém matrizes que podem conter números complexos. Isso ocorre se o sistema tiver polos complexos conjugados, de modo que o estado final pode não ser estritamente real. Isso gera um problema, pois algoritmos que normalmente trabalham com LMI não estão aptos a lidar com sistemas com esta situação. Nesta seção, foi apresentada uma forma de lidar com isso [40, 41].

Seja o sistema governado pela seguinte FT, no qual os polos sejam números complexos conjugados:

$$G(s) = \frac{\alpha + \beta i}{s + b + \omega i} + \frac{\alpha - \beta i}{s + b - \omega i} \quad (20)$$

A resposta a um degrau unitário para este sistema pode ser escrita como:

$$S(k\Delta t) = d^0 + d^d e^{rk\Delta t} + \bar{d}^d e^{\bar{r}k\Delta t} \quad (21)$$

Assim:

$$d^0 = \frac{2(\alpha b + \omega \beta)}{b^2 + \omega^2}$$

$$d^d = \alpha + \beta i$$

$$\bar{d}^d = \alpha - \beta i$$

$$r = -b - \omega i$$

$$\bar{r} = -b + \omega i$$

Mas (21) pode ser desenvolvida da seguinte forma:

$$\begin{aligned} S(k\Delta t) &= d^0 + (\alpha + \beta i)(\cos(\omega k\Delta t) - i\sin(\omega k\Delta t))e^{-bk\Delta t} + \\ &\quad (\alpha - \beta i)(\cos(\omega k\Delta t) + i\sin(\omega k\Delta t))e^{-bk\Delta t} \Leftrightarrow \\ S(k\Delta t) &= d^0 + e^{-bk\Delta t}(2\alpha\cos(\omega k\Delta t) + 2\beta\sin(\omega k\Delta t)) \end{aligned} \quad (22)$$

Desta forma, a matriz de estados seria representada por:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-b\Delta t} (\cos(\omega\Delta t) + i\sin(\omega k\Delta t)) & 0 \\ 0 & 0 & e^{-b\Delta t} (\cos(\omega\Delta t) - i\sin(\omega k\Delta t)) \end{bmatrix} \quad (23)$$

Tal matriz tem os mesmos autovalores da seguinte matriz, que tem somente estados reais (que é o que se observa na equação (22)):

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-b\Delta t} \cos(\omega\Delta t) & -e^{-b\Delta t} \sin(\omega\Delta t) \\ 0 & e^{-b\Delta t} \sin(\omega\Delta t) & e^{-b\Delta t} \cos(\omega\Delta t) \end{bmatrix} \quad (24)$$

Assim, a segunda matriz pode substituir a primeira na representação do estado, desde que a resposta obtida com o degrau seja equivalente àquela obtida em (21). Para, isso, deve-se calcular os valores para as matrizes $\mathbf{B}$ e $\mathbf{C}$. No caso de $\mathbf{B}$, os valores corretos são obtidos substituindo-se os valores da diagonal principal de $\mathbf{D}^d$ por:

$$\mathbf{D}^d = \begin{bmatrix} d_1^d & 0 \\ 0 & \bar{d}_2^d \end{bmatrix} = \begin{bmatrix} \alpha - \beta & \alpha + \beta \\ -\alpha - \beta & \alpha - \beta \end{bmatrix} \quad (25)$$

Assim, a matriz $\mathbf{B}$ é dada por:

$$\mathbf{B} = \begin{bmatrix} d^0 \\ 2e^{(-b\Delta t)} (\alpha \cos(\omega\Delta t) + \beta \sin(\omega\Delta t)) \\ 2e^{(-b\Delta t)} (\beta \cos(\omega\Delta t) + \alpha \sin(\omega\Delta t)) \end{bmatrix} \quad (26)$$

No caso de $\mathbf{C}$, é necessário modificar os valores de Ψ para não computar os estados mais de uma vez. É feita a seguinte substituição:

$$\Psi = \begin{bmatrix} 1 & 1 \end{bmatrix} \Rightarrow \Psi = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (27)$$

Deste modo, $\mathbf{C}$ passa a ser dada por:

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \quad (28)$$

Uma generalização para sistemas maiores pode ser encontrada em Santoro [40].

2.2.3 OPOM para sistemas com tempo morto

Nesta seção, discute-se a adaptação do OPOM para operar com tempo morto. González e Odloak [26] propuseram uma primeira abordagem para tanto. Como esta resultava em um modelo que nem sempre era detectável, Santoro e Odloak [21] propuseram outro modelo, que também foi utilizado por Martins e Odloak [23] em seu controlador robusto.

Tal sistema consiste em trazer para as variáveis de estado parte das ações de controle anteriores que têm relevância somente no instante presente, devido ao tempo morto. Seja um MIMO cujo maior tempo morto seja dado por $\theta_{\max}$. Este seria então descrito por:

$$\begin{aligned} \begin{bmatrix} \mathbf{x}^s(k+1) \\ \mathbf{x}^d(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{I}_{ny} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} + \begin{bmatrix} \mathbf{x}^s(k) \\ \mathbf{x}^d(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_0^s \\ \mathbf{B}_0^d \end{bmatrix} \Delta \mathbf{u}(k) + \\ &\quad \begin{bmatrix} \mathbf{B}_1^s \\ \mathbf{B}_1^d \end{bmatrix} \Delta \mathbf{u}(k-1) + \dots + \begin{bmatrix} \mathbf{B}_{\theta_{\max}}^s \\ \mathbf{B}_{\theta_{\max}}^d \end{bmatrix} \Delta \mathbf{u}(k-\theta_{\max}) \end{aligned} \quad (29)$$

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{I}_{ny} & \mathbf{\Psi} \end{bmatrix} \mathbf{x}(k) \quad (30)$$

Rearranjando as equações (29) e (30) para se adequarem ao formato tradicional de estado conforme (8) e (9), obtém-se:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}^s(k) \\ \mathbf{x}^d(k) \\ \Delta \mathbf{u}(k-1) \\ \Delta \mathbf{u}(k-2) \\ \vdots \\ \Delta \mathbf{u}(k-\theta_{\max}) \end{bmatrix} \quad (31) \quad \mathbf{A} = \begin{bmatrix} \mathbf{I}_{ny} & \mathbf{0} & \mathbf{B}_1^s & \mathbf{B}_2^s & \cdots & \mathbf{B}_{\theta_{\max}-1}^s & \mathbf{B}_{\theta_{\max}}^s \\ \mathbf{0} & \mathbf{F} & \mathbf{B}_1^d & \mathbf{B}_2^d & \cdots & \mathbf{B}_{\theta_{\max}-1}^d & \mathbf{B}_{\theta_{\max}}^d \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{nu} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{nu} & \mathbf{0} \end{bmatrix} \quad (32)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_0^s \\ \mathbf{B}_0^d \\ \mathbf{I}_{nu} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (33)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{ny} & \mathbf{\Psi} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \quad (34)$$

Os detalhes quanto à criação das matrizes $\mathbf{B}^s$ e $\mathbf{B}^d$ podem ser vistos em Santoro e Odloak [21].

2.3 Controlabilidade, alcançabilidade, observabilidade e detectabilidade

Um determinado sistema linear é controlável se, desde qualquer estado inicial é possível obter uma sequência de controle de modo que a origem seja alcançada em um tempo finito. Este mesmo sistema será alcançável se for possível, desde a origem obter uma sequência de controle de modo que um estado final arbitrário seja atingido em um tempo finito. Controlabilidade não necessariamente implica em alcançabilidade, pois se o estado inicial for $\mathbf{x}(k) = \mathbf{0}$ (k é o instante inicial), o sistema é controlável com a sequência de controle $[\mathbf{u}^T(k) \cdots \mathbf{u}^T(k+m-1)]^T = \mathbf{0}$, mesmo que não seja alcançável [41, 42].

Considerando o sistema representado em (1) e (2) e assumindo que o problema parta de um estado inicial $\mathbf{x}(k)$, no instante n (sendo n a ordem do sistema), o estado é dado por:

$$\mathbf{x}(k+n) = \mathbf{A}^n \mathbf{x}(k) + \mathbf{A}^{n-1} \mathbf{B} \mathbf{u}(k) + \cdots + \mathbf{B} \mathbf{u}(k+n-1) \quad (35)$$

Matricialmente:

$$\mathbf{x}(k+n) = \mathbf{A}^n \mathbf{x}(k) + \mathbf{W}_c \mathbf{U} \quad (36)$$

Em que:

$$\mathbf{W}_c = [\mathbf{B} \ \mathbf{A}\mathbf{B} \ \cdots \ \mathbf{A}^{n-1}\mathbf{B}] \quad (37)$$

$$\mathbf{U} = [\mathbf{u}^T(k+n-1) \ \cdots \ \mathbf{u}^T(k)]^T \quad (38)$$

Define-se $\mathbf{W}_c$ a matriz de controlabilidade. Se esta apresentar um posto igual a n , é possível obter um sistema de n equações lineares e definir a sequência de controle $\mathbf{U}$ que leve o sistema do estado inicial para qualquer estado desejado. Se um estado inicial estiver localizado na origem e a matriz de controlabilidade apresentar posto pleno, este sistema é dito alcançável [42, 43].

Na literatura de sistemas lineares, define-se como observável um sistema se, para um estado inicial $\mathbf{x}(k)$, existir um horizonte de tempo finito $k+l$ dado que, a partir das entradas e saídas no intervalo $[k, k+l]$, o estado inicial possa ser determinado. Mais uma vez, considerando o sistema representado em (1) e (2) com as saídas conhecidas e, por uma questão de simplificação, com todas as entradas nulas, então:

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \mathbf{x}(k) = \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \\ \vdots \\ \mathbf{y}(k+n-1) \end{bmatrix} \quad (39)$$

Define-se $\mathbf{W}_o$ a matriz de observabilidade:

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \quad (40)$$

Caso esta matriz apresente posto n , o sistema é dito observável e o estado $\mathbf{x}(k)$ pode ser obtido através de (39).

Um conceito mais abrangente do que o de observabilidade é o de detectabilidade. Nele, um sistema é detectável se for observável, ou se os seus estados não observáveis tenderem a zero. Define-se o critério de Popov-Belevitch-Hautus para detectabilidade como: um sistema é detectável se, dado λ , então a matriz $\begin{bmatrix} \mathbf{A} - \lambda \mathbf{I} \\ \mathbf{C} \end{bmatrix}$, tal que $|\lambda| > 1$, tem posto pleno.

A importância de mostrar que o sistema é alcançável está intimamente ligada à definição do controlador. Um determinado set-point pode ser ou não atingível a depender da alcançabilidade do sistema. Já a importância da detectabilidade é que, caso o sistema seja detectável, garante-se a existência de um observador de estados, o que permite que as estimativas converjam para os valores reais, garantindo portanto o controle do sistema.

2.4 Observadores de estado e o filtro de Kalman

Além dos modelos em espaço de estados, cabe falar dos observadores de estado. Como os modelos são meras aproximações da realidade, eles estão sujeitos a diversos tipos de erros, sendo os seguintes os principais: aproximações (o sistema pode não ser linear em toda sua faixa de operação), estimação de parâmetros das matrizes $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ incorretos, erros de medidas intrínsecos aos medidores utilizados e distúrbios desconhecidos [44, 45]. Assim, é necessário utilizar-se de um método para fazer o modelo convergir para o resultado da planta real. Os sistemas que realizam esta tarefa são chamados de observadores ou estimadores de estado. Estes foram desenvolvidos para garantir as rotas de espaçonaves nas décadas de 1950 e 60. Um dos marcos nesta teoria foi o desenvolvimento do Filtro de Kalman, o mais conhecido dentre os estimadores e hoje utilizado em diversas áreas do conhecimento e em tecnologias como GPS etc. [46].

Para que se compare o estado previsto por um modelo de um sistema com o estado real, é necessário que o estado seja detectável, ou seja, que se possa obter seu valor através dos valores de entradas e saídas do sistema [42]. Considerando novamente o sistema representado em (1) e (2) e assumindo que as matrizes $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ representem a planta exatamente, o objetivo passa a ser calcular os valores de $\mathbf{x}(k)$ baseando-se nas entradas $\mathbf{u}(k)$, $\mathbf{u}(k-1)$ etc. e saídas $\mathbf{y}(k)$, $\mathbf{y}(k-1)$ etc. Tal cálculo pode ser feito de modo direto [41], ou com observadores de estado.

Como notação nas subseções seguintes, considere o símbolo $\hat{\cdot}$ (circunflexo) acima de qualquer variável como estimativa, assim, $\hat{\mathbf{x}}(k)$ é o estado estimado; o símbolo $\tilde{\cdot}$ (til) abaixo

como erro de uma estimativa em relação ao sistema real, de forma que, $\tilde{\mathbf{x}}(k)$ é o erro entre o estado estimado e o estado real; qualquer variável sem símbolo é o valor medido do sistema, ou seja, $\mathbf{x}(k)$ é o estado medido. Desta forma, $\tilde{\mathbf{x}}(k) = \hat{\mathbf{x}}(k) - \mathbf{x}(k)$.

2.4.1 Obtenção do estado através de estimadores

Considere a estimativa do estado através do modelo:

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) \quad (41)$$

Se o sistema for assintoticamente estável, o modelo nominal for exatamente o mesmo da planta, não houver distúrbios e os erros de medida forem negligenciáveis, $\hat{\mathbf{x}}$ tende a $\mathbf{x}$. Para garantir a convergência quando estas hipóteses não são respeitadas, pode-se adicionar o erro de previsão relativo à medida das saídas $\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k)$ [45]:

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}[\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k)] \quad (42)$$

$\mathbf{L}$ é chamada a matriz de ganho do observador. Sendo o erro de medida $\tilde{\mathbf{x}}(k)$, então, da equação (42):

$$\tilde{\mathbf{x}}(k+1) = (\mathbf{A} - \mathbf{L}\mathbf{C})\tilde{\mathbf{x}}(k) \quad (43)$$

Estando os auto-valores de $(\mathbf{A} - \mathbf{L}\mathbf{C})$ dentro do círculo unitário, o erro tende a zero assintoticamente e a estimativa do estado tende ao valor real. A existência de uma matriz $\mathbf{L}$ que permita que estes auto-valores estejam dentro do círculo unitário é garantida caso o sistema seja detectável [42].

2.4.2 O filtro de Kalman

Resultado do trabalho de Rudolf Kálmán, o filtro de Kalman é um observador de estado que trabalha, como descrito na seção anterior, como um estimador. Nele, assume-se a existência de ruídos aleatórios nas medidas e nos modelos. Desta forma pode-se considerar o filtro de Kalman como um “observador estocástico” [47].

Tanto o estado inicial, o modelo e as medidas estão sujeitos a erros aleatórios e têm uma distribuição (considerada gaussiana, de média nula, com os ruídos brancos e não-correlacionados). Se novas medidas não forem realizadas ao longo do tempo, o fato de o modelo ser sujeito a erros faz com que o estado previsto tenha sua variância aumentada em relação ao instante anterior. Entretanto, a combinação da nova medida no tempo com a previsão do estado neste mesmo instante reduz a variância da estimativa, fazendo com que esta tenda ao valor real [47]. Como as medidas são então repetidas sucessivamente, o filtro de Kalman é chamado de recursivo e diz-se que ele é implementado “online”.

Seu modelo pode ser descrito pela equação a seguir, na qual $\mathbf{w}(k)$ é o ruído no modelo e $\mathbf{v}(k)$ é o ruído na medida das saídas. Por hipótese, assume-se neste trabalho que $\mathbf{w}(k)$ e $\mathbf{v}(k)$ são descorrelacionados.

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \quad (44)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \quad (45)$$

O filtro de Kalman é considerado um estimador ótimo, pois a obtenção da matriz $\mathbf{L}$ e a minimização dos erros é feita através de um método de mínimos quadrados [47]. Sejam $\mathbf{R}_1$ a matriz de covariância do erro no modelo e $\mathbf{R}_2$ a matriz de covariância do erro nas medidas. Seja $\mathbf{P}$ a matriz de covariância do erro de estimação. Então, o ganho do filtro é calculado minimizando-se o valor esperado do erro da estimativa da seguinte forma [45]:

$$\min E \left[(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) \times (\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T \right] \quad (46)$$

Para um tempo suficientemente grande, o ganho do filtro é dado por:

$$\mathbf{L} = (\mathbf{A}\mathbf{P}\mathbf{C}^T) \times (\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}_2)^{-1} \quad (47)$$

Em que:

$$\mathbf{P} = \mathbf{A}\mathbf{P}\mathbf{A}^T + \mathbf{R}_1 - (\mathbf{A}\mathbf{P}\mathbf{C}^T) \times (\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}_2)^{-1} \times (\mathbf{A}\mathbf{P}\mathbf{C}^T)^T \quad (48)$$

Posteriormente ao filtro de Kalman, que assume um modelo de estado linear e um erro quadrático, houve desenvolvimentos no sentido de se utilizar modelos mais complexos, visando diminuir ainda mais os erros de medida. Tais métodos são normalmente chamados de filtros de Kalman estendidos [48]. Uma possível desvantagem é que estes modelos mais complexos não são necessariamente convexos, já que não partem de modelos lineares, de modo que o mínimo obtido nem sempre é um mínimo global. Houve pesquisadores inclusive que questionaram as estimativas feitas pelo filtro de Kalman estendido. Um exemplo famoso é dado em [49], mas que força situações iniciais esdrúxulas para que o filtro falhe.

2.5 O controlador preditivo baseado em modelo (MPC)

O controlador preditivo baseado em modelo é uma estratégia de controle na qual a ação de controle é obtida ao minimizar-se, a cada período de amostragem, uma função chamada “função objetivo” calculada a partir das predições de trajetória do sistema em malha aberta. O valor que esta função assume a cada instante, por sua vez, é chamado de “custo”. O problema de otimização consiste na minimização da função objetivo associada a certas restrições, como por exemplo nas entradas e na movimentação das entradas. A abordagem tradicional é minimizar a diferença entre a trajetória prevista para as saídas e um valor desejado, comumente chamado de “set-point” [20].

O processo a ser controlado pode ser representado por um modelo como descrito na seção 2.2, mais especificamente, pelas equações (1) e (2), ou então, se utilizados incrementos nas entradas, pelas equações (8) e (9). Seja m o horizonte de controle, ou seja, o horizonte no qual ações de controle são aplicadas e q o horizonte de predição das trajetórias nas saídas, então, uma função objetivo típica será representada pela seguinte função quadrática:

$$V_k = \sum_{j=1}^q \left\| \mathbf{y}(k+j|k) - \mathbf{y}_k^{sp} \right\|_{\mathbf{Q}}^2 + \sum_{j=0}^{m-1} \left\| \Delta \mathbf{u}(k+j|k) \right\|_{\mathbf{R}}^2 \quad (49)$$

Na qual $\mathbf{y}_k^{sp}$ é o set-point das saídas, $\Delta \mathbf{u}$ é a variação nas ações de controle, dada por: $\Delta \mathbf{u}(k+j|k) = \mathbf{u}(k+j|k) - \mathbf{u}(k+j-1|k)$, $\mathbf{u}$ são as entradas e $\mathbf{Q}$ e $\mathbf{R}$ são as matrizes de peso nas saídas e penalização nas ações de controle das entradas, respectivamente. Quanto às restrições para o problema de otimização, estas podem assumir diversas formas. São comuns as expressões a seguir:

$$\mathbf{u}^{min} \leq \mathbf{u}(k+j|k) \leq \mathbf{u}^{max}, \quad j = 0, \dots, m-1 \quad (50)$$

$$-\Delta \mathbf{u}^{max} \leq \Delta \mathbf{u}(k+j|k) \leq \Delta \mathbf{u}^{max}, \quad j = 0, \dots, m-1 \quad (51)$$

$$\Delta \mathbf{u}(k+j|k) = \mathbf{0}, \quad j \geq m \quad (52)$$

Pode-se modificar a função objetivo para considerar um horizonte infinito de predição das trajetórias nas saídas. Este tem a vantagem de apresentar estabilidade garantida [15, 16]. Com isso, pode-se reescrever a função objetivo como:

$$V_k = \sum_{j=1}^{\infty} \left\| \mathbf{y}(k+j|k) - \mathbf{y}_k^{sp} \right\|_{\mathbf{Q}}^2 + \sum_{j=0}^{\infty} \left\| \Delta \mathbf{u}(k+j|k) \right\|_{\mathbf{R}}^2 \quad (53)$$

Quanto às restrições, valem neste caso também (50) - (52). A equação (52) garante que a segunda soma infinita passe a ser finita: $\sum_{j=0}^{m-1} \left\| \Delta \mathbf{u}(k+j|k) \right\|_{\mathbf{R}}^2$.

Para que a primeira soma infinita passe a ser finita, é necessário que $\lim_{j \rightarrow \infty} \mathbf{A}^j = \mathbf{0}$. Isso deve ocorrer porque, graças a (52), após o horizonte de controle m , a equação de estado (8) se escreve: $\mathbf{x}(k+m+1) = \mathbf{A}\mathbf{x}(k+m)$. Dessa forma, se o limite acima tender a zero, a soma infinita tende para um valor finito. Assim, pode reescrever-se a soma: $\sum_{j=m+1}^{\infty} \left\| \mathbf{y}(k+j|k) - \mathbf{y}_k^{sp} \right\|_{\mathbf{Q}}^2 = \left\| \mathbf{C}\mathbf{x}(k+m|k) - \mathbf{y}_k^{sp} \right\|_{\bar{\mathbf{Q}}}^2$, na qual $\bar{\mathbf{Q}}$ é a chamada matriz de custo terminal e pode ser obtida através de equação de Lyapunov: $\bar{\mathbf{Q}} - \mathbf{A}^T \bar{\mathbf{Q}} \mathbf{A} = \mathbf{A}^T \mathbf{Q} \mathbf{A}$.

Finalmente, é comum também adicionar uma variável de folga (em inglês: *slack*) ao problema. Seja δ_k a variável de folga, então a equação (49) pode ser modificada da seguinte forma:

$$V_k = \sum_{j=1}^q \left\| \mathbf{y}(k+j|k) - \mathbf{y}_k^{sp} - \delta_k \right\|_{\mathbf{Q}}^2 + \sum_{j=0}^{m-1} \left\| \Delta \mathbf{u}(k+j|k) \right\|_{\mathbf{R}}^2 + \left\| \delta_k \right\|_{\mathbf{S}}^2 \quad (54)$$

Sujeita às restrições (50) - (52), como anteriormente (já que as folgas são irrestritas). Neste caso, $\mathbf{S}$ é a matriz de penalização do uso das variáveis de folga e deve ter ordem de

grandeza significativamente maior que $\mathbf{R}$ para garantir que não haja uso indiscriminado de tais variáveis pelo controlador.

2.5.1 Convexidade das funções objetivo e dos conjuntos gerados pelas restrições

Seja a definição de conjunto convexo dada por: um conjunto $\mathcal{S} \subset \mathbb{R}^n$ é dito convexo se a combinação convexa de dois elementos pertencentes ao conjunto também estiver contida neste [50]. Em outras palavras, se $\mathbf{x}_1$ e $\mathbf{x}_2$ pertencem a $\mathcal{S}$, então $\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2$ (que é a combinação convexa destas variáveis) deve estar contido em $\mathcal{S}$ para $\lambda \in [0, 1]$. Por indução, pode-se considerar qualquer média ponderada da forma $\sum_{j=1}^k \lambda_j \mathbf{x}_j$ na qual $\sum_{j=1}^k \lambda_j = 1$ e $\lambda_j \geq 0$ para $j = 1, \dots, k$ como uma combinação convexa de $\mathbf{x}_1$ a $\mathbf{x}_k$. Se é permitida a escolha de qualquer $\lambda_j \in \mathbb{R}$, a combinação é dita linear.

Dito isto, restrições do tipo (50) e (51) formam conjuntos convexos em $\mathbb{R}^{nu}$. Estes conjuntos podem ser denominados $\mathcal{U}^{nu}$. Tome por exemplo valores quaisquer $\mathbf{u}_1$ e $\mathbf{u}_2$ que obedeçam à restrição (50), ou seja, pertençam a $\mathcal{U}^{nu}$. Assim, para λ tal que $\lambda \in [0, 1]$:

$$\begin{aligned} \begin{cases} \mathbf{u}^{min} \leq \mathbf{u}_1 \leq \mathbf{u}^{max} \\ \mathbf{u}^{min} \leq \mathbf{u}_2 \leq \mathbf{u}^{max} \end{cases} &\Rightarrow \begin{cases} \lambda \mathbf{u}^{min} \leq \lambda \mathbf{u}_1 \leq \lambda \mathbf{u}^{max} \\ (1 - \lambda) \mathbf{u}^{min} \leq (1 - \lambda) \mathbf{u}_2 \leq (1 - \lambda) \mathbf{u}^{max} \end{cases} \Rightarrow \\ &\mathbf{u}^{min} \leq \lambda \mathbf{u}_1 + (1 - \lambda) \mathbf{u}_2 \leq \mathbf{u}^{max} \end{aligned}$$

Fica fácil observar que $\lambda \mathbf{u}_1 + (1 - \lambda) \mathbf{u}_2 \in \mathcal{U}^{nu}$. Como isto é válido para quaisquer pares de entradas, pela definição acima, $\mathcal{U}^{nu}$ é um conjunto convexo.

Seja agora a seguinte definição de função convexa: dada $f : \mathcal{S} \rightarrow \mathbb{R}$, em que $\mathcal{S}$ é um conjunto convexo em $\mathbb{R}^n$ conforme definido acima, f é dita convexa em $\mathcal{S}$ se [50]:

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2) \quad (55)$$

Na equação acima, $\mathbf{x}_1$ e $\mathbf{x}_2$ são quaisquer elementos de $\mathcal{S}$ e λ é qualquer número real tal que $\lambda \in [0, 1]$. Uma propriedade importante de funções convexas é que a soma de funções

convexas também é uma função convexa. Ou seja, $f(\mathbf{x}) = \sum_{j=1}^n \alpha_j f_j(\mathbf{x})$ para $\alpha_j > 0$ e $j = 1, \dots, n$.

Outra definição, ligada à concavidade da função f é a da matriz Hessiana de f . Seja f duas vezes diferenciável e com as segundas derivadas parciais contínuas, então a matriz $n \times n$ simétrica $\mathbf{H}$ é dada por:

$$\mathbf{H}(f(\mathbf{x})) = \begin{bmatrix} f_{11}(\mathbf{x}) & \cdots & f_{1n}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ f_{n1}(\mathbf{x}) & \cdots & f_{nn}(\mathbf{x}) \end{bmatrix} \quad (56)$$

Na qual $f_{ij}(\mathbf{x}) = \partial f(\mathbf{x}) / \partial x_i \partial x_j$. Assim, uma condição suficiente (mas não necessária) para que f seja convexa é se a matriz Hessiana for positiva e semi-definida para cada ponto em $\mathcal{S}$.

Finalmente, como uma aplicação da teoria de convexidade para a otimização, considere-se o problema de minimizar a função $f(\mathbf{x})$ sujeita a $\mathbf{x} \in \mathcal{S}$. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, então, qualquer ponto $\mathbf{x} \in \mathcal{S}$ é dito uma solução factível para o problema. Se $\mathbf{x}^* \in \mathcal{S}$ é tal que $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ para qualquer $\mathbf{x} \in \mathcal{S}$, então $\mathbf{x}^*$ é dita uma solução ótima ou simplesmente uma solução para o problema. Se f for uma função convexa e $\mathcal{S}$ for um conjunto convexo, então $\mathbf{x}^*$ é dita uma solução ótima global para o problema e é única [50].

Dadas estas definições, fica fácil observar que funções quadráticas do tipo $f(\mathbf{x}) = \|\mathbf{x}\|_{\mathbf{Q}}^2$, nas quais $\mathbf{Q}$ é uma matriz diagonal positiva e definida são convexas. Uma forma simples de mostrar isso é observar que a matriz Hessiana de tais funções será $\mathbf{H}(\mathbf{x}) = 2\mathbf{Q}$, que é positiva e definida para qualquer $\mathbf{x}$.

Dessa forma, lembrando que a soma de funções convexas também é convexa e lembrando também que as restrições (50) e (51) formam conjuntos convexas, decorre que as funções objetivo descritas pelas equações (49), (53) e (54) são convexas e que existe uma única sequência de ações de controle $\mathbf{u}^*$ que minimiza o problema de otimização a cada instante k .

2.6 O MPC em uma estrutura de camadas de otimização

Em plantas modernas, MPCs centralizados são normalmente implementados como parte de uma hierarquia de controle multinível [51, 52]. A camada superior, muitas vezes conhecida como “Otimização em tempo real” (em inglês: *Real Time Optimization* (RTO)), utiliza modelos rigorosos e prevê condições de operação ótimas em estado estacionário, normalmente visando à maximização do lucro. Tais condições são enviadas para a camada do MPC que move a planta de um estado estacionário para o próximo, alterando os set-points para o nível de PID.

Diversos trabalhos visando integrar RTOs a MPCs já foram desenvolvidos. Inicialmente, observaram-se tentativas de integrar diversas camadas, como feito em Farina *et al.* [33], em que um gerador de referências é utilizado. Long *et al.* [53] utilizam um MPC central, que distribui informações aos subsistemas. Mais recentemente, montou-se um sistema análogo ao da Figura 1, que pode ser visto em Li e Swartz [54].

Figura 1: Estrutura de controle: Integração RTO/MPC

Fonte: adaptado de González e Odloak (2009) [25].

Em muitos processos, a meta da camada de MPC não é guiar todas as variáveis controladas para set-points ou valores desejados, mas simplesmente mantê-las dentro de faixas apropriadas. Isso é conhecido como controle por faixas, ou em inglês, “zone control” [25, 55], de modo que o ponto ótimo de operação pode ser definido através de alvos ou “tar-

gets” nas variáveis manipuladas. Tal esquema de operação é interessante por possibilitar trabalhar com sistemas não quadrados, com mais saídas do que entradas. Como há uma faixa de operação para cada saída, em muitos casos, distúrbios não levam os valores destas variáveis para fora de suas faixas. Assim, não se gera ação de controle por parte do controlador, permitindo que as variáveis manipuladas se concentrem em ações que levem as variáveis controladas que estão fora de suas faixas novamente para dentro das mesmas. Um arranjo comum de funcionamento deste tipo de integração pode ser visto na Figura 1. Para outras estruturas de integração com otimização, vide Scatollini [32].

Como se pode observar na estrutura da Figura 1, a camada de otimização gera valores denominados $\mathbf{u}_k^{des}$, estes são chamados de alvos para as entradas. A partir daí, o MPC atua utilizando seu modelo linear de modo a gerar a atuação de controle a ser enviada para o sistema. Não há alvos nas saídas (embora uma estrutura que mescale alvos nas entradas e nas saídas pode ser desenvolvida também - vide Martin, Zanin e Odloak [19]), estas são controladas entre faixas contendo um limite máximo e mínimo ($\mathbf{y}^{max}$, $\mathbf{y}^{min}$).

Dada a estrutura definida no parágrafo anterior e na Figura 1, o problema de otimização é dado por [25]:

$$\min_{\Delta \mathbf{u}, \mathbf{y}_k^{sp}, \boldsymbol{\delta}_k^y, \boldsymbol{\delta}_k^u} V_k \quad (57)$$

$$\mathbf{u}^{min} \leq \mathbf{u}(k+j|k) \leq \mathbf{u}^{max}, \quad j = 0, \dots, m-1 \quad (58)$$

$$-\Delta \mathbf{u}^{max} \leq \Delta \mathbf{u}(k+j|k) \leq \Delta \mathbf{u}^{max}, \quad j = 0, \dots, m-1 \quad (59)$$

$$\mathbf{y}^{min} \leq \mathbf{y}_k^{sp} \leq \mathbf{y}^{max} \quad (60)$$

$$\Delta \mathbf{u}(k+j|k) = \mathbf{0}, \quad j \geq m \quad (61)$$

$$\mathbf{y}(k+m|k) - \mathbf{y}_k^{sp} - \boldsymbol{\delta}_k^y = \mathbf{0} \quad (62)$$

$$\mathbf{u}(k+m|k) - \mathbf{u}_k^{des} - \boldsymbol{\delta}_k^u = \mathbf{0} \quad (63)$$

Em que:

$$\begin{aligned} V_k = & \sum_{j=1}^{\infty} \|\mathbf{y}(k+j|k) - \mathbf{y}_k^{sp} - \boldsymbol{\delta}_k^y\|_{\mathbf{Q}_y}^2 + \sum_{j=0}^{\infty} \|\mathbf{u}(k+j|k) - \mathbf{u}_k^{des} - \boldsymbol{\delta}_k^u\|_{\mathbf{Q}_u}^2 \\ & + \sum_{j=1}^m \|\Delta \mathbf{u}(k+j|k)\|_{\mathbf{R}}^2 + \|\boldsymbol{\delta}_k^y\|_{\mathbf{S}_y}^2 + \|\boldsymbol{\delta}_k^u\|_{\mathbf{S}_u}^2 \end{aligned} \quad (64)$$

Repare que na função objetivo há uma diferença entre o valor de entrada e o valor desejado $\mathbf{u}(k+j|k) - \mathbf{u}_k^{des}$ para ser minimizado. Repare também que há um set-point para as saídas, $\mathbf{y}_k^{sp}$, mas este valor deixa de ser fixo e passa a ser uma variável. Além disso, há variáveis de folga nas saídas e também nas entradas, garantindo que haja solução mesmo no caso de o alvo não ser factível. Adicionalmente, há uma matriz de pesos $\mathbf{Q}_u$ para as entradas (em contraste a $\mathbf{Q}_y$, matriz de pesos para as saídas). Escolhendo os valores destas duas matrizes em relação uma a outra, pode-se priorizar a otimização (busca do alvo nas entradas) aumentando os valores de $\mathbf{Q}_u$ ou a garantia de operação dentro de faixas (busca de manter as saídas dentro dos limites mínimo e máximo) aumentando os valores de $\mathbf{Q}_y$. Finalmente, adicionam-se matrizes de penalização para o uso das folgas $\mathbf{S}_y$ e $\mathbf{S}_u$.

Quanto às restrições para o problema, estas assumem a forma de inequações (58) - (60), como para o controlador definido na seção 2.5, e equações (61) - (63) para os custos terminais associados às somas infinitas.

Repare que, como $\mathbf{y}_k^{sp}$ é agora variável, deve estar restrita aos valores das faixas de operação nas saídas. Repare também que se as variáveis de folga não forem devidamente penalizadas, o controlador pode utilizá-las excessivamente para satisfazer (62) e (63) e prejudicar a estratégia de controle.

2.6.1 O MPC para sistemas com tempo morto

Dada a adaptação do OPOM para sistemas com tempo morto, conforme descrito na subseção 2.2.3, é possível adaptar o MPC em uma estrutura de camadas com RTO para operar com sistemas com tempo morto.

González e Odloak [26] abordam o tema inicialmente. A seguir Santoro e Odloak [21] propõem um controlador para operar em sistemas com tempo morto (que também é capaz de lidar com polos integradores). Posteriormente, Martins e Odloak [23] adaptam a mesma formulação para um controlador robusto.

Tal formulação consiste em supor uma camada de otimização superior única que fornecerá alvos (ou “targets” em inglês) para a camada do MPC. O modelo utilizado é o OPOM proposto na subseção 2.2.3. A função objetivo a ser minimizada é a mesma descrita em (64). Lembre que $\theta_{\max}$ é o maior tempo morto dentro do sistema controlado. Assim, para

que a soma infinita da função objetivo seja restrita, a restrição (62) deve ser adaptada para:

$$\mathbf{y}(k + m + \theta_{\max}|k) - \mathbf{y}_k^{sp} - \boldsymbol{\delta}_k^y = \mathbf{0} \quad (65)$$

Dessa forma, o problema adaptado para a situação com tempo morto passa a ser minimizar (57) sujeita a (58) - (61), (63) e (65) e o modelo da subseção 2.2.3.

2.7 O controlador preditivo baseado em modelo distribuído (DMPC)

2.7.1 Diferenças entre os modelos de estado para diferentes tipos de sistema

Como já mencionado na seção 2.1, existem basicamente três subdivisões de sistemas com relação ao grau de centralização esperado: os centralizados, os distribuídos e os descentralizados. Para os modelos descentralizados não se tem mais somente uma matriz $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ para todo o sistema, como em (1) e (2), mas sim um conjunto de matrizes para cada subsistema. Isto pode ser representado, para M subsistemas, da seguinte forma [31]:

$$\mathbf{x}_i(k+1) = \mathbf{A}_{ii}\mathbf{x}_i(k) + \mathbf{B}_{ii}\mathbf{u}_i(k) \quad i = 1, \dots, M \quad (66)$$

$$\mathbf{y}_i(k) = \mathbf{C}_i\mathbf{x}_i(k) \quad i = 1, \dots, M \quad (67)$$

Neste esquema, cada conjunto $\mathbf{u}_i$ de variáveis manipuladas (entradas), com dimensão nu_i , controla um conjunto $\mathbf{y}_i$ de variáveis controladas (saídas), com dimensão ny_i . O estado $\mathbf{x}_i$, por sua vez, tem dimensão nx_i . Venkat *et al.* [31] chamam este arranjo de subsistemas de “modelo particionado”. De fato, todos os valores possíveis para cada $\mathbf{u}_i$ e $\mathbf{y}_i$, $i = 1, \dots, M$ formam conjuntos disjuntos, não vazios cuja união é o modelo para o sistema centralizado. Se comparadas com o modelo centralizado, as matrizes $\mathbf{A}_{ii}$, $\mathbf{B}_{ii}$ e $\mathbf{C}_i$ constituem a diagonal principal das matrizes $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$.

Para um sistema qualquer:

$$\begin{bmatrix} \mathbf{x}_1(k+1) \\ \vdots \\ \mathbf{x}_i(k+1) \\ \vdots \\ \mathbf{x}_M(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1i} & \cdots & \mathbf{A}_{1M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{A}_{i1} & \cdots & \mathbf{A}_{ii} & \cdots & \mathbf{A}_{iM} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{A}_{M1} & \cdots & \mathbf{A}_{Mi} & \cdots & \mathbf{A}_{MM} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(k) \\ \vdots \\ \mathbf{x}_i(k) \\ \vdots \\ \mathbf{x}_M(k) \end{bmatrix} + \\
 \begin{bmatrix} \mathbf{B}_{11} & \cdots & \mathbf{B}_{1i} & \cdots & \mathbf{B}_{1M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{B}_{i1} & \cdots & \mathbf{B}_{ii} & \cdots & \mathbf{B}_{iM} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{B}_{M1} & \cdots & \mathbf{B}_{Mi} & \cdots & \mathbf{B}_{MM} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1(k) \\ \vdots \\ \mathbf{u}_i(k) \\ \vdots \\ \mathbf{u}_M(k) \end{bmatrix} \quad (68)$$

$$\begin{bmatrix} \mathbf{y}_1(k) \\ \vdots \\ \mathbf{y}_i(k) \\ \vdots \\ \mathbf{y}_M(k) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1 & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{C}_i & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{C}_M \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(k) \\ \vdots \\ \mathbf{x}_i(k) \\ \vdots \\ \mathbf{x}_M(k) \end{bmatrix} \quad (69)$$

Em que:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_i \\ \vdots \\ \mathbf{u}_M \end{bmatrix} \quad (70)$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_i \\ \vdots \\ \mathbf{x}_M \end{bmatrix} \quad (71)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_i \\ \vdots \\ \mathbf{y}_M \end{bmatrix} \quad (72)$$

Como um modelo é constituído a priori pelas matrizes $\mathbf{A}_{ii}$, $\mathbf{A}_{ij}$, $\mathbf{B}_{ii}$, $\mathbf{B}_{ij}$ e $\mathbf{C}_i$ mas as matrizes $\mathbf{A}_{ij}$ e $\mathbf{B}_{ij}$ são negligenciadas em modelos descentralizados, claramente não se leva em conta nestes casos a interação entre um subsistema i e um subsistema j , de modo que tal situação só pode ser aplicada caso as interações entre diferentes sistemas sejam negligenciáveis.

Já em sistemas distribuídos, a estratégia de controle consiste em dividir o todo em subsistemas do mesmo modo que anteriormente, porém incluir nos M modelos as matrizes (i, j) , de modo que o sistema i conheça as implicações de uma ação de controle tomada

pelos outros sistemas j e vice-versa. Isso é feito, da seguinte forma:

$$\mathbf{x}_i(k+1) = \sum_{j=1}^M [\mathbf{A}_{ij}\mathbf{x}_j(k) + \mathbf{B}_{ij}\mathbf{u}_j(k)] \quad i = 1, \dots, M \quad (73)$$

$$\mathbf{y}_i(k) = \mathbf{C}_i\mathbf{x}_i(k) \quad i = 1, \dots, M \quad (74)$$

2.7.2 Diferenças entre controladores para cada grau de centralização

Assim como os sistemas, os controladores também se subdividem em centralizado, descentralizado e distribuído. Além disso, um controlador distribuído pode ser de dois tipos: cooperativo ou não-cooperativo.

Um controlador centralizado utiliza equações de estado para sistemas centralizados, como (1) e (2) ou (8) e (9). Um controlador descentralizado utiliza equações de estado para sistemas descentralizados, (66) e (67). Um controlador distribuído, por sua vez, utiliza equações de estado para sistemas distribuídos, como (73) e (74). Em controladores descentralizados ou distribuídos, restrições do tipo $\mathbf{u}_i^{min} \leq \mathbf{u}_i \leq \mathbf{u}_i^{max}$ também são aplicadas para as entradas de cada subsistema. Estas costumam ser as fronteiras para os conjuntos $\mathcal{U}^{nu_i}$, de modo que $\mathbf{u}_i(k) \in \mathcal{U}^{nu_i}$, $i = 1, \dots, M$.

Além das restrições, a função objetivo determina o tipo de estratégia de controle. Em um controlador centralizado, a função objetivo V_k calculada no instante k é comum a todos os controladores e assume formas do tipo das equações (49), (53) ou (54). Considerando-se um horizonte de controle m , o problema de otimização resolvido pelo controlador é, genericamente:

$$\min_{\mathbf{u}_k} V_k(\mathbf{x}, \mathbf{u}) \quad (75a)$$

$$\mathbf{u}(l|k) \in \mathcal{U}^{nu} \quad l = k, \dots, k+m-1 \quad (75b)$$

Em que o estado é dado pelas equações (1) e (2) e as restrições sumarizadas por (75b) podem ser escritas na forma de (50) - (52). Além disso, nas expressões acima, $\mathbf{u}_k$, que é a variável do problema, é o vetor de entradas ao longo do horizonte de predição:

$$\mathbf{u}_k = \begin{bmatrix} \mathbf{u}(k|k) \\ \vdots \\ \mathbf{u}(k+m-1|k) \end{bmatrix} \quad (76)$$

Tal problema calcula a solução ótima para todo o sistema ao mesmo tempo, de modo que possíveis conflitos entre as variáveis sejam resolvidos de maneira ótima. Em um controlador descentralizado, por sua vez, cada subsistema resolve um problema de otimização individualmente, com uma função objetivo e restrições independentes:

$$\min_{\mathbf{u}_{ki}} V_{ki}(\mathbf{x}_i, \mathbf{u}_i) \quad i = 1, \dots, M \quad (77a)$$

$$\mathbf{u}_i(l|k) \in \mathcal{U}^{nu_i} \quad l = k, \dots, k+m-1, \quad i = 1, \dots, M \quad (77b)$$

V_{ki} está aqui genericamente mencionada, mas ela também assume a mesma forma quadrática das equações (49), (53) ou (54), só que para cada subsistema i . O estado é dado por (66) e (67) e a variável $\mathbf{u}_{ki}$ é definida como:

$$\mathbf{u}_{ki} = \begin{bmatrix} \mathbf{u}_i(k|k) \\ \vdots \\ \mathbf{u}_i(k+m-1|k) \end{bmatrix} \quad (78)$$

Como mencionado na subseção 2.7.1, este arranjo só é possível se as interações entre um subsistema e outro forem negligenciáveis.

Chegando agora ao sistema distribuído não-cooperativo, cada controlador minimiza uma função objetivo distinta, como em (77a) e age sob as restrições (77b), porém recebe informações dos outros sistemas através das equações de estado (73) e (74).

$$\min_{\mathbf{u}_{ki}} V_{ki}(\mathbf{x}_j, \mathbf{u}_j) \quad i, j = 1, \dots, M \quad (79a)$$

$$\mathbf{u}_i(l|k) \in \mathcal{U}^{nu_i} \quad l = k, \dots, k+m-1, \quad i = 1, \dots, M \quad (79b)$$

Um ponto importante a se acrescentar é que tal arranjo deve ser feito de modo iterativo, pois, para o subsistema i , as informações ótimas de outros subsistemas j só estão disponíveis até o instante anterior $k-1$. Assim, modifica-se a equação (73) e escreve-se o estado

como:

$$\mathbf{x}_i(k+1) = \mathbf{A}_{ii}\mathbf{x}_i(k) + \mathbf{B}_{ii}\mathbf{u}_i(k) + \sum_{\substack{j=1 \\ j \neq i}}^M \left[\mathbf{A}_{ij}\mathbf{x}_j^{(p-1)}(k) + \mathbf{B}_{ij}\mathbf{u}_j^{(p-1)}(k) \right] \quad i = 1, \dots, M \quad (80)$$

Na equação (80), p representa a iteração atual e $p-1$ representa a iteração anterior. Assim as informações $\mathbf{u}_j^{(p-1)}$ são referentes à iteração anterior à atual. Na primeira iteração ($p = 1$), pode-se escolher, por exemplo, $\mathbf{x}_j^{(0)}(k) = \mathbf{x}_j(k-1)$ e $\mathbf{u}_j^{(0)}(k) = \mathbf{u}_j(k-1)$. Desta forma, é possível lidar com subsistemas que apresentem alguma interação entre si.

Tal arranjo não-cooperativo pode levar a um equilíbrio de Nash, o que pode ser prejudicial para o controle do sistema. Daí vem a ideia de definir um novo tipo de sistema, o chamado cooperativo. Nele, cada controlador i manipula somente as variáveis $\mathbf{u}_i$, como nos controladores descentralizado e distribuído não-cooperativo, porém todos minimizam uma função comum, o que deve levar a um ótimo de Pareto.

Em linhas gerais, um equilíbrio de Nash é definido em teoria dos jogos como aquele em que, dado um número de jogadores e um número de possíveis escolhas dentro do jogo, cada jogador faz sua escolha dadas as escolhas dos outros jogadores e, a partir daí, não tem mais nada a ganhar mudando de escolha. Um ótimo de Pareto, por sua vez, é aquele obtido quando todos os jogadores têm um objetivo comum ou uma estratégia conjunta baseada em um acordo prévio. Para detalhes acerca de teoria dos jogos e definições matemáticas do equilíbrio de Nash e do ótimo de Pareto, vide Osborne e Rubinstein [56] e Başar e Olsder [57].

O controlador distribuído e cooperativo é então genericamente definido como:

$$\min_{\mathbf{u}_{ki}} V_k(\mathbf{x}, \mathbf{u}) \quad i = 1, \dots, M \quad (81a)$$

$$\mathbf{u}_i(l|k) \in \mathcal{U}^{nu_i} \quad l = k, \dots, k+m-1, \quad i = 1, \dots, M \quad (81b)$$

A forma mais comum de definir V_k nestes casos, por sua vez, é como uma combinação convexa de cada função objetivo:

$$V_k(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^M w_i V_{ki}(\mathbf{x}_j, \mathbf{u}_j) \quad j = 1, \dots, M \quad (82a)$$

$$\sum_{i=1}^M w_i = 1 \quad w_i \geq 0 \quad \forall i \quad (82b)$$

Assim como no caso anterior, cada controlador i utiliza somente as variáveis $\mathbf{u}_i$ para minimizar a função e recebe informação das variáveis $\mathbf{u}_j$ referentes à iteração anterior. A diferença, porém, é que ao minimizar uma função comum, cada controlador leva em conta de maneira mais eficiente o efeito que sua atuação tem nos outros subsistemas [31].

Como se pôde observar, o MPC centralizado foi amplamente desenvolvido e modificado, incluindo características como otimização em camadas, tempo morto, possibilidade de lidar com faixas ao invés de set-points etc. Para o caso distribuído entretanto, não há tanta ênfase nestes desenvolvimentos na literatura. O foco maior é em sistemas quadrados com set-points nas saídas e o estudo da convergência por modelos iterativos. Incrementos nas entradas são pouco utilizados e variáveis de folga são praticamente inexistentes. Questões como RTO e tempo morto são somente citadas como possibilidades. Desta forma, o presente trabalho buscou preencher estas lacunas utilizando arranjos e modelos de convergência já previamente estudados e adaptando os mesmos a trabalhar com variáveis de folga, incrementos nas entradas, etc.

3 Modelos em espaço de estados para sistemas distribuídos

Uma abordagem genérica de modelos em espaço de estados já foi dada na seção 2.2. Nessa mesma seção discutiu-se detalhadamente um destes modelos, o OPOM e suas derivações. Na subseção 2.7.1 foram abordados, também de forma genérica, os modelos de estado para sistemas distribuídos e sua relação com os modelos centralizados. Coube agora, neste capítulo, definir os modelos de estado utilizados no presente trabalho. Estes foram derivados do OPOM e adaptados para o trabalho com sistemas distribuídos. Inicialmente, definiu-se o OPOM para sistemas distribuídos (seção 3.1), na sequência, foi elaborada uma adaptação do OPOM para sistemas com tempo morto (seção 3.2). Finalmente, discutiram-se as questões de controlabilidade, alcançabilidade etc. para modelos distribuídos (seção 3.3).

3.1 OPOM adaptado para sistemas distribuídos

Como já mencionado na subseção 2.7.1, modelos distribuídos são normalmente definidos como subdivisões (ou então modelos particionados) em relação a modelos para sistemas centralizados. Assim, a maneira mais óbvia de definir o modelo utilizado neste trabalho foi tomar as matrizes para o OPOM ($\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$) e subdividi-las nas matrizes $\mathbf{A}_{ij}$, $\mathbf{B}_{ij}$ e $\mathbf{C}_{ij}$.

Conforme definido na subseção 2.2.1, a matriz $\mathbf{A}$ é diagonal para sistemas estáveis em malha aberta. Assim, olhando as equações (10) e (68), percebe-se imediatamente que $\mathbf{A}_{ij} = \mathbf{0}$, de modo que são as matrizes $\mathbf{B}_{ij}$ que carregam as informações de ações de controle executadas pelos subsistemas j para o subsistema i e suas implicações. Traduzindo isto para o formato do OPOM:

$$\begin{bmatrix} \mathbf{x}_i^s(k+1) \\ \mathbf{x}_i^d(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^s(k) \\ \mathbf{x}_i^d(k) \end{bmatrix} + \begin{bmatrix} \mathbf{D}_{ii}^0 \\ \mathbf{D}_i^d \mathbf{F}_i \mathbf{N}_{ii} \end{bmatrix} \Delta \mathbf{u}_i(k) +$$

$$\sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \begin{bmatrix} \mathbf{D}_{ij}^0 \\ \mathbf{D}_j^d \mathbf{F}_j \mathbf{N}_{ij} \end{bmatrix} \Delta \mathbf{u}_j(k) \right\} \quad (83)$$

$$\mathbf{y}_i(k) = \begin{bmatrix} \mathbf{I}_i & \mathbf{\Psi}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^s(k) \\ \mathbf{x}_i^d(k) \end{bmatrix} \quad (84)$$

As dimensões das matrizes são, por sua vez: $\mathbf{D}_{ii}^0 \in \mathbb{R}^{ny_i \times nu_i}$; $\mathbf{F}_i \in \mathbb{R}^{nd_i \times nd_i}$, em que $nd_i = ny_i \cdot nu_i \cdot na_i$ e na_i é o maior grau entre os denominadores das funções de transferência pertencentes ao subsistema i ; $\mathbf{D}_i^d \in \mathbb{R}^{nd_i \times nd_i}$; $\mathbf{J}_{ij} \in \mathbb{R}^{nu_i \cdot na_i \times nu_i}$; $\mathbf{N}_{ij} \in \mathbb{R}^{nd_i \times nu_i}$; $\mathbf{\Phi}_i \in \mathbb{R}^{nu_i \cdot na_i}$ e $\mathbf{\Psi}_i \in \mathbb{R}^{ny_i \times nd_i}$ (já pertencentes ao domínio dos números reais como descrito na subseção 2.2.2).

Como exemplo, para ilustrar a concepção destas matrizes, seja um sistema 3×2 , $(y_1, y_2, y_3, u_1, u_2)$ subdividido em dois subsistemas (y_1, y_2, u_1) e (y_3, u_3) , cujas FTs tenham todas $na = 1$ e sejam genericamente dadas por:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \\ G_{31} & G_{32} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

As matrizes $\mathbf{A}$, $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{B}$, $\mathbf{D}_{11}^0$, $\mathbf{D}_{12}^0$, $\mathbf{D}_{21}^0$, $\mathbf{D}_{22}^0$, $\mathbf{D}_1^d$, $\mathbf{D}_2^d$, $\mathbf{N}_{11}$, $\mathbf{N}_{12}$, $\mathbf{N}_{21}$, $\mathbf{N}_{22}$, $\mathbf{C}$, $\mathbf{\Psi}_1$ e $\mathbf{\Psi}_2$ para o sistema centralizado encontram-se abaixo, suas componentes que compõem cada subsistema estão destacadas por cores: vermelho para o subsistema (y_1, y_2, u_1) e azul para o subsistema (y_3, u_3) .

$$\mathbf{A} = \begin{bmatrix} \textcolor{red}{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \textcolor{red}{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \textcolor{blue}{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{r_{11}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{r_{12}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{r_{21}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{r_{22}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{r_{31}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{r_{32}} & 0 \end{bmatrix}$$

$$\mathbf{F}_1 = \begin{bmatrix} \textcolor{red}{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \textcolor{red}{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{r_{11}} & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{r_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{r_{21}} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{r_{22}} \end{bmatrix}$$

$$\mathbf{F}_2 = \begin{bmatrix} \textcolor{blue}{1} & 0 & 0 \\ 0 & e^{r_{31}} & 0 \\ 0 & 0 & e^{r_{32}} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} d_{11}^0 & d_{12}^0 \\ d_{21}^0 & d_{22}^0 \\ d_{31}^0 & d_{32}^0 \\ d_1^d e^{r_{11}} & 0 \\ 0 & d_2^d e^{r_{12}} \\ d_3^d e^{r_{21}} & 0 \\ 0 & d_4^d e^{r_{22}} \\ d_5^d e^{r_{31}} & 0 \\ 0 & d_6^d e^{r_{32}} \end{bmatrix}$$

$$\mathbf{D}_{11}^0 = \begin{bmatrix} d_{11}^0 \\ d_{21}^0 \end{bmatrix}$$

$$\mathbf{D}_{12}^0 = \begin{bmatrix} d_{12}^0 \\ d_{22}^0 \end{bmatrix}$$

$$\mathbf{D}_{21}^0 = \begin{bmatrix} d_{31}^0 \end{bmatrix}$$

$$\mathbf{D}_{22}^0 = \begin{bmatrix} d_{32}^0 \end{bmatrix}$$

$$\mathbf{D}_1^d = \begin{bmatrix} d_1^d & 0 & 0 & 0 \\ 0 & d_2^d & 0 & 0 \\ 0 & 0 & d_3^d & 0 \\ 0 & 0 & 0 & d_4^d \end{bmatrix}$$

$$\mathbf{D}_2^d = \begin{bmatrix} d_5^d & 0 \\ 0 & d_6^d \end{bmatrix}$$

$$\mathbf{N}_{11} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{N}_{12} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{N}_{21} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{N}_{22} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\Psi_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\Psi_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

3.2 OPOM adaptado para sistemas distribuídos com tempo morto

Da mesma forma que se fez uma divisão em vários subsistemas para o caso sem tempo morto, vale ainda o esquema de divisão em subsistemas dado por (73) e (74) em M subsistemas. Esse está adaptado para se ajustar ao esquema (29) - (34). Em (32), a matriz $\mathbf{A}$ não é mais diagonal, no entanto, os vários valores de $\mathbf{B}^s$ e $\mathbf{B}^d$ também podem ser subdivididos de modo que não haja relação entre $\mathbf{x}_i$ e $\mathbf{x}_j$, $i \neq j$.

Em um determinado subsistema i , o estado receberá a ação de controle das variáveis $\Delta \mathbf{u}_i$, entretanto, estarão contidas no vetor de estado também as entradas $\Delta \mathbf{u}_j$, $j \neq i$ para os instantes anteriores. Isso deve ser feito, já que uma ação de controle no sistema j afeta o sistema i e vice-versa.

$$\begin{aligned} \mathbf{x}_i(k) = & \left[\mathbf{x}_i^{sT}(k) \quad \mathbf{x}_i^{dT}(k) \quad \Delta \mathbf{u}_1^T(k-1) \cdots \Delta \mathbf{u}_i^T(k-1) \cdots \Delta \mathbf{u}_M^T(k-1) \right. \\ & \Delta \mathbf{u}_1^T(k-2) \cdots \Delta \mathbf{u}_i^T(k-2) \cdots \Delta \mathbf{u}_M^T(k-2) \cdots \Delta \mathbf{u}_1^T(k-\theta_{\max}) \\ & \left. \cdots \Delta \mathbf{u}_i^T(k-\theta_{\max}) \cdots \Delta \mathbf{u}_M^T(k-\theta_{\max}) \right]^T \end{aligned} \quad (85)$$

$$\mathbf{A}_{ii} = \begin{bmatrix} \mathbf{I}_i & \mathbf{0} & \mathbf{B}_1^s & \mathbf{B}_2^s & \cdots & \mathbf{B}_{\theta_{\max}-1}^s & \mathbf{B}_{\theta_{\max}}^s \\ \mathbf{0} & F_i & \mathbf{B}_1^d & \mathbf{B}_2^d & \cdots & \mathbf{B}_{\theta_{\max}-1}^d & \mathbf{B}_{\theta_{\max}}^d \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{nu} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{nu} & \mathbf{0} \end{bmatrix} \quad (86)$$

A criação das matrizes $\mathbf{B}^s$ e $\mathbf{B}^d$ obedece à mesma lógica do caso centralizado, no entanto, as mesmas vão se reagrupar da seguinte forma: o primeiro subscrito mostra o retorno no tempo, assim $\mathbf{B}_1^s$ é referente ao instante $k-1$; os dois subscritos na sequência e entre parêntesis (ij) referem-se à influência da entrada j no subsistema i . Dessa forma:

$$\begin{aligned} \mathbf{B}_1^s &= \left[\mathbf{B}_{1(11)}^s \cdots \mathbf{B}_{1(1i)}^s \cdots \mathbf{B}_{1(1M)}^s \cdots \mathbf{B}_{1(M1)}^s \cdots \mathbf{B}_{1(Mi)}^s \cdots \mathbf{B}_{1(MM)}^s \right] \\ \mathbf{B}_2^s &= \left[\mathbf{B}_{2(11)}^s \cdots \mathbf{B}_{2(1i)}^s \cdots \mathbf{B}_{2(1M)}^s \cdots \mathbf{B}_{2(M1)}^s \cdots \mathbf{B}_{2(Mi)}^s \cdots \mathbf{B}_{2(MM)}^s \right] \\ &\vdots \\ \mathbf{B}_{\theta_{\max}}^s &= \left[\mathbf{B}_{\theta_{\max}(11)}^s \cdots \mathbf{B}_{\theta_{\max}(1i)}^s \cdots \mathbf{B}_{\theta_{\max}(1M)}^s \cdots \mathbf{B}_{\theta_{\max}(M1)}^s \cdots \mathbf{B}_{\theta_{\max}(Mi)}^s \cdots \mathbf{B}_{\theta_{\max}(MM)}^s \right] \end{aligned}$$

$$\mathbf{B}_{ii} = \begin{bmatrix} \mathbf{B}_{0(ii)}^s \\ \mathbf{B}_{0(ii)}^d \\ \mathbf{I}_i \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (87)$$

$$\mathbf{B}_{ij} = \begin{bmatrix} \mathbf{B}_{0(ij)}^s \\ \mathbf{B}_{0(ij)}^d \\ \mathbf{0}_i \\ \mathbf{I}_j \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (88)$$

$$\mathbf{C}_i = \begin{bmatrix} \mathbf{I}_i & \Psi_i & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \quad (89)$$

Repare que em $\mathbf{A}_{ii}$, as matrizes identidade têm a dimensão de todas as entradas nu , e não somente da parte distribuída nu_i . Como mencionado anteriormente, todas as entradas de tempos de amostragem anteriores devem fazer parte do estado de um determinado subsistema, já que todas as entradas potencialmente interferem no subsistema, mesmo aquelas que não manipulam diretamente o mesmo.

3.3 Controlabilidade, alcançabilidade, observabilidade e detectabilidade dos modelos distribuídos

Aqui, valem as mesmas premissas e definições dadas na seção 2.3. Considere entretanto o sistema representado em (73) e (74) com as matrizes $\mathbf{A}_{ij}$ nulas, como é de fato no OPOM distribuído definido nas seções anteriores. Então, assumindo que o problema seja dividido em M subsistemas e que um subsistema qualquer i parta de um estado inicial $\mathbf{x}_i(k)$, no instante n_i (sendo n_i a ordem do sistema), o estado é dado por:

$$\begin{aligned} \mathbf{x}_i(k + n_i) = & \mathbf{A}_{ii}^{n_i} \mathbf{x}_i(k) + \mathbf{A}_{ii}^{n_i-1} \mathbf{B}_{ii} \mathbf{u}_i(k) + \cdots + \mathbf{B}_{ii} \mathbf{u}_i(k + n_i - 1) + \\ & \sum_{\substack{j=1 \\ j \neq i}}^M \{ \mathbf{A}_{ii}^{n_i-1} \mathbf{B}_{ij} \mathbf{u}_j(k) + \cdots + \mathbf{B}_{ij} \mathbf{u}_j(k + n_i - 1) \} \quad i = 1, \dots, M \end{aligned} \quad (90)$$

Matricialmente:

$$\mathbf{x}_i(k + n_i) = \mathbf{A}_{ii}^{n_i} \mathbf{x}(k) + \mathbf{W}_{cii} \mathbf{U}_i + \sum_{\substack{j=1 \\ j \neq i}}^M \{ \mathbf{W}_{cij} \mathbf{U}_j \} \quad i = 1, \dots, M \quad (91)$$

Na qual:

$$\mathbf{W}_{cii} = [\mathbf{B}_{ii} \quad \mathbf{A}_{ii} \mathbf{B}_{ii} \quad \cdots \quad \mathbf{A}_{ii}^{n_i-1} \mathbf{B}_{ii}] \quad i = 1, \dots, M \quad (92)$$

$$\mathbf{W}_{cij} = [\mathbf{B}_{ij} \quad \mathbf{A}_{ii} \mathbf{B}_{ij} \quad \cdots \quad \mathbf{A}_{ii}^{n_i-1} \mathbf{B}_{ij}] \quad i, j = 1, \dots, M \quad (93)$$

$$\mathbf{U}_i = [\mathbf{u}_i^T(k + n_i - 1) \quad \cdots \quad \mathbf{u}_i^T(k)]^T \quad i = 1, \dots, M \quad (94)$$

Se $\mathbf{W}_{cii}$ apresentar posto igual a $M \times n_i$, é possível definir a sequência de controle $\mathbf{U}_i$ que leva o sistema do estado inicial para o desejado e este sistema é dito alcançável. Cabe lembrar que $\mathbf{U}_i$ é a variável enquanto que $\mathbf{U}_j$ são parâmetros para o controlador i .

Com relação à observabilidade e mais uma vez considerando o sistema representado em (73) e (74) com as matrizes $\mathbf{A}_{ij}$ nulas, com as saídas conhecidas e, por uma questão de simplificação, com todas as entradas nulas, então:

$$\begin{bmatrix} \mathbf{C}_i \\ \mathbf{C}_i \mathbf{A}_{ii} \\ \vdots \\ \mathbf{C}_i \mathbf{A}_{ii}^{n_i-1} \end{bmatrix} \mathbf{x}_i(k) = \begin{bmatrix} \mathbf{y}_i(k) \\ \mathbf{y}_i(k+1) \\ \vdots \\ \mathbf{y}_i(k+n_i-1) \end{bmatrix} \quad i = 1, \dots, M \quad (95)$$

Define-se $\mathbf{W}_{oi}$ a matriz de observabilidade para cada subsistema:

$$\mathbf{W}_{oi} = \begin{bmatrix} \mathbf{C}_i \\ \mathbf{C}_i \mathbf{A}_{ii} \\ \vdots \\ \mathbf{C}_i \mathbf{A}_{ii}^{n_i-1} \end{bmatrix} \quad i = 1, \dots, M \quad (96)$$

Caso esta matriz apresente posto n_i , $\forall i$, o subsistema é dito observável e o estado $\mathbf{x}_i(0)$ pode ser obtido através de (95). A detectabilidade é definida analogamente ao caso centralizado: para $i = 1, \dots, M$, dado λ_i um autovalor de $\mathbf{A}_{ii}$ e $Re(\lambda_i) \geq 0$, então as matrizes $\begin{bmatrix} \mathbf{A}_{ii} - \lambda_i \mathbf{I}_i \\ \mathbf{C}_i \end{bmatrix}$ têm posto pleno. Assim como no caso centralizado, a importância de mostrar que o sistema é alcançável está intimamente ligada à definição do controlador. Um determinado set-point pode ser ou não atingível a depender da alcançabilidade do sistema. Já a importância da detectabilidade é que, caso o sistema seja detectável, garante-se a existência de um observador de estados para cada subsistema.

O OPOM para sistemas distribuídos é detectável, assim como seu análogo centralizado. Assim, neste trabalho, o filtro de Kalman foi utilizado como observador de estados para todos os controladores definidos em cada subsistema. Utilizaram-se nas equações (46) - (48) as matrizes $\mathbf{A}_{ii}$ e $\mathbf{C}_i$ e considerou-se $\mathbf{R}_1 = 0.5\mathbf{I}_{nx_i}$ e $\mathbf{R}_2 = 0.5\mathbf{I}_{nx_i}$. Consideraram-se como medidas apenas as saídas $\mathbf{y}_i$, e não variáveis de estado $\mathbf{x}_i$. Finalmente, trabalhou-se com o modelo nominal.

4 Proposta para o controlador distribuído: DC-IHMPC

Uma vez definidos os modelos em espaço de estados para os controladores distribuídos a partir do OPOM, chegou o momento de definir os controladores. Neste capítulo foram discutidas, em detalhes, a função objetivo e as restrições para a mesma do primeiro controlador desenvolvido neste trabalho. Este utiliza o modelo definido na seção 3.1, trabalha com “set-points” nas saídas e em sistemas com igual número de entradas e saídas.

Inicialmente, foi formulada a função objetivo e as restrições para o controlador distribuído e cooperativo (seção 4.1). Parte importante desta seção é a subseção 4.1.1, na qual se discute o modelo iterativo sob o qual os controladores dos diversos subsistemas devem atuar para convergir para a solução ótima.

A seguir, uma discussão matemática sobre factibilidade recursiva e estabilidade foi realizada na seção 4.2. Nesta mesma seção, também foi dada a prova matemática da convergência do modelo iterativo (subseção 4.2.2). Finalmente, resultados de simulação para sistemas simplificados também são apresentados neste capítulo (seção 4.3). Tais simulações têm por objetivo mostrar algumas características do controlador e permitir discussões acerca do mesmo.

4.1 Formulação da função objetivo e das restrições para o DC-IHMPC

Como já mencionado na subseção 2.7.2, no caso do controlador distribuído e cooperativo, convém definir V_k como uma combinação convexa de funções genericamente denominadas V_{ki} , como elucidado nas equações (82a) e (82b). Cabe agora mostrar o problema de otimização, definir em detalhe quem são as funções V_{ki} e também quais são exatamente as restrições sob as quais o sistema funciona. O controlador foi intitulado “*Distributed and Cooperative Infinite Horizon Model Predictive Control*” ou simplesmente DC-IHMPC, pois trabalha de modo cooperativo e distribuído e usa horizonte infinito de predição.

Cada função V_{ki} para cada subsistema $i = 1, \dots, M$ é definida analogamente à função objetivo de um sistema centralizado utilizada por Odloak [20], utilizando-se um

MPC de horizonte infinito, de modo a ter convergência garantida, diferentemente de Rawlings e Stewart [1] e Stewart *et al.* [30], que utilizam uma função da forma: $V_{ki} = \sum_{l=1}^m \left\{ \|y(k+l|k)\|_{\mathbf{Q}_i}^2 + \|u(k+l|k)\|_{\mathbf{R}_i}^2 \right\}$, $\mathbf{Q}_i, \mathbf{R}_i > \mathbf{0}$. Assim, para o presente trabalho:

$$V_{ki} = \sum_{l=1}^{\infty} \left\| \mathbf{y}_i(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} \right\|_{\mathbf{Q}_i}^2 + \sum_{l=0}^{m-1} \left\| \Delta \mathbf{u}_i(k+l|k) \right\|_{\mathbf{R}_i}^2 + \sum_{j=1}^M \left\| \boldsymbol{\delta}_{kij} \right\|_{\mathbf{S}_i}^2 \quad (97)$$

Lembre que m é o horizonte de controle. Observe também que V_{ki} contém uma variável de folga $\boldsymbol{\delta}_{kii}$. Quanto a $\boldsymbol{\delta}_{kij}$, $j \neq i$, estes serão parâmetros para o controlador i e variáveis para os outros controladores. As matrizes $\mathbf{Q}_i$, $\mathbf{R}_i$ e $\mathbf{S}_i$ são matrizes positivas diagonais de pesos para cada variável. As matrizes $\mathbf{Q}_i$ priorizam qual saída tem maior peso, as matrizes $\mathbf{R}_i$ priorizam qual entrada tem seus movimentos mais penalizados pela função objetivo e as matrizes $\mathbf{S}_i$, que são a penalização da variável de folga devem ser selecionadas com valores mais altos que $\mathbf{Q}_i$ e $\mathbf{R}_i$, de modo que o controlador tente sempre zerar a parcela $\left\| \boldsymbol{\delta}_{kii} \right\|_{\mathbf{S}_i}^2$, deixando de utilizar a folga. A necessidade da mesma ficará aparente conforme as restrições para o problema forem discutidas a seguir.

O parâmetro $\mathbf{y}_{ki}^{sp}$ representa o set-point para a saída i no instante k , de modo que na equação (97), a rigor, é feita a minimização do erro quadrático entre a saída e seu set-point. Finalmente, como já observado antes, o controlador é definido em função de variações nas entradas, utilizando-se os modelos definidos no Capítulo 3.

A soma infinita da equação (97), por sua vez, pode ser dividida em duas partes e reescrita como:

$$\sum_{l=1}^{\infty} \left\| \mathbf{y}_i(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} \right\|_{\mathbf{Q}_i}^2 = \sum_{l=1}^m \left\| \mathbf{y}_i(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} \right\|_{\mathbf{Q}_i}^2 + \sum_{l=m+1}^{\infty} \left\| \mathbf{y}_i(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} \right\|_{\mathbf{Q}_i}^2 \quad (98)$$

A primeira parte da soma é minimizada normalmente como uma função quadrática. À segunda parte da soma, indo de $m+1$ até ∞ , dá-se o nome de custo termina. Esta parte

precisa convergir para um valor finito para garantir que o controlador também convirja.

Lembre que, do OPOM adaptado para sistemas distribuídos (equação (84)): $\mathbf{y}_i(k) = \mathbf{x}_i^s(k) + \Psi_i \mathbf{x}_i^d(k)$. Assim, a segunda parte se reescreve como:

$$\begin{aligned} \sum_{l=m+1}^{\infty} \left\| \mathbf{y}_i(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} \right\|_{\mathbf{Q}_i}^2 = \\ \sum_{l=m+1}^{\infty} \left\| \mathbf{x}_i^s(k+l|k) + \Psi_i \mathbf{x}_i^d(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} \right\|_{\mathbf{Q}_i}^2 \end{aligned} \quad (99)$$

Para instantes posteriores ao horizonte de controle, $\Delta \mathbf{u}_i = \mathbf{0}$, de modo que $\mathbf{x}_i^s$ passa a ter um valor fixo. Além disso, $\mathbf{y}_{ki}^{sp}$ é constante. Desta forma, se estes valores forem somados infinitas vezes, para garantir que esta soma seja finita, é necessário que $\mathbf{x}_i^s - \mathbf{y}_{ki}^{sp} = \mathbf{0}$ para todos os instantes $l \geq m$. Assim é necessário adicionar-se a seguinte restrição:

$$\mathbf{x}_i^s(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} = \mathbf{0} \quad l \geq m \quad (100)$$

Aqui fica clara a necessidade da variável de folga: caso os set-points em todos os subsistemas sejam alcançáveis, o controlador não utiliza as variáveis de folga, estas passam a ser nulas e a diferença entre o set-point e $\mathbf{x}_i^s$ passa a ser zero para $l \geq m$, caso contrário, as folgas são escolhidas pelo controlador de modo a serem iguais à diferença entre o set-point e $\mathbf{x}_i^s$, garantindo a restrição da equação (100). Combinando-se (99) e (100), chega-se a seguinte equação:

$$\begin{aligned} \sum_{l=m+1}^{\infty} \left\| \mathbf{x}_i^s(k+l|k) + \Psi_i \mathbf{x}_i^d(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} \right\|_{\mathbf{Q}_i}^2 = \\ \sum_{l=1}^{\infty} \left\| \Psi_i \mathbf{x}_i^d(k+m+l|k) \right\|_{\mathbf{Q}_i}^2 \end{aligned} \quad (101)$$

Já $\mathbf{x}_i^d(k+l|k) = \mathbf{F}_i^{l-m} \mathbf{x}_i^d(k+m|k)$, $l \geq m+1$, então:

$$\sum_{l=1}^{\infty} \left\| \Psi_i \mathbf{x}_i^d(k+m+l|k) \right\|_{\mathbf{Q}_i}^2 = \sum_{l=1}^{\infty} \left\| \Psi_i \mathbf{F}_i^l \mathbf{x}_i^d(k+m|k) \right\|_{\mathbf{Q}_i}^2 \quad (102)$$

Como $\lim_{l \rightarrow \infty} \mathbf{F}_i^l = \mathbf{0}$ (polos estáveis dentro do círculo unitário), a soma infinita converge para um valor finito. Desta forma, o “custo terminal” adquire a forma de uma equação de Lyapunov, de modo que (102) se escreve da seguinte forma:

$$\sum_{l=1}^{\infty} \left\| \Psi_i \mathbf{x}_i^d(k+m+l|k) \right\|_{\mathbf{Q}_i}^2 = \left\| \mathbf{x}_i^d(k+m|k) \right\|_{\bar{\mathbf{Q}}_i}^2 \quad (103a)$$

$$\bar{\mathbf{Q}}_i - \left\| \mathbf{F}_i \right\|_{\bar{\mathbf{Q}}_i}^2 = \left\| \mathbf{F}_i \Psi_i \right\|_{\mathbf{Q}_i}^2 \quad (103b)$$

Ou seja, V_{ki} assume a seguinte expressão:

$$\begin{aligned} V_{ki} = & \sum_{l=1}^m \left\| \mathbf{y}_i(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kij} \right\|_{\mathbf{Q}_i}^2 + \left\| \mathbf{x}_i^d(k+m|k) \right\|_{\bar{\mathbf{Q}}_i}^2 + \\ & \sum_{l=0}^{m-1} \left\| \Delta \mathbf{u}_i(k+l|k) \right\|_{\mathbf{R}_i}^2 + \sum_{j=1}^M \left\| \boldsymbol{\delta}_{kij} \right\|_{\mathbf{S}_i}^2 \end{aligned} \quad (104)$$

Cabe lembrar que a restrição (100) representa na verdade M equações, pois a função objetivo será V_k e não V_{ki} . além disso, como $\mathbf{x}_i^s$ não varia após m , basta que a restrição seja válida em m . Levando estas considerações em conta, para o controlador i , pode-se reescrever a restrição (100) como:

$$\mathbf{x}_q^s(k+m|k) - \mathbf{y}_{kq}^{sp} - \boldsymbol{\delta}_{kqi} - \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kqj} = \mathbf{0} \quad q = 1, \dots, M \quad (105)$$

Com isso, fica fácil notar que cada controlador tem M variáveis de folga. Para o controlador i , $\boldsymbol{\delta}_{kqi}$, $q = 1, \dots, M$ são variáveis, enquanto que $\boldsymbol{\delta}_{kqj}$, $j \neq i$ são parâmetros. Por uma questão de notação, pode-se agrupar as folgas em um único vetor:

$$\bar{\delta}_{ki} = \begin{bmatrix} \delta_{k1i} \\ \delta_{k2i} \\ \vdots \\ \delta_{kMi} \end{bmatrix} \quad i = 1, \dots, M \quad (106)$$

Seria interessante, agora, reescrever a soma da equação (104) em uma forma matricial utilizando-se as matrizes oriundas do OPOM, de modo que se possa mais facilmente programar sua solução como programação quadrática (em inglês: *Quadratic Programing* (QP)). Para isso, seja:

$$\Delta \mathbf{u}_{ki} = \begin{bmatrix} \Delta \mathbf{u}_i(k|k) \\ \vdots \\ \Delta \mathbf{u}_i(k+m-1|k) \end{bmatrix} \quad (107)$$

Sejam também as matrizes $\mathbf{D}_{mij}^0$, $\bar{\mathbf{I}}_i$, $\mathbf{F}_{xi}$, $\mathbf{F}_{fui}$, $\mathbf{F}_{duij}$ e $\mathbf{F}_{uij}$:

$$\mathbf{D}_{mij}^0 = \begin{bmatrix} \mathbf{D}_{ij}^0 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{D}_{ij}^0 & \cdots & \mathbf{D}_{ij}^0 \end{bmatrix} \quad (108)$$

$$\bar{\mathbf{I}}_i = \begin{bmatrix} \mathbf{I}_{ny_i} \\ \vdots \\ \mathbf{I}_{ny_i} \end{bmatrix} \quad (109)$$

$$\mathbf{F}_{xi} = \begin{bmatrix} \mathbf{F}_i \\ \mathbf{F}_i^2 \\ \vdots \\ \mathbf{F}_i^m \end{bmatrix} \quad (110)$$

$$\mathbf{F}_{fui} = \begin{bmatrix} \mathbf{I}_{nx_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{F}_i & \mathbf{I}_{nx_i} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_i^{m-1} & \mathbf{F}_i^{m-2} & \cdots & \mathbf{I}_{nx_i} \end{bmatrix} \quad (111)$$

$$\mathbf{F}_{duij} = \begin{bmatrix} \mathbf{D}_i^d \mathbf{N}_{ij} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_i^d \mathbf{N}_{ij} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_i^d \mathbf{N}_{ij} \end{bmatrix} \quad (112)$$

$$\mathbf{F}_{uij} = \mathbf{F}_{fui} \times \mathbf{F}_{duij} \quad (113)$$

Sejam, ainda:

$$\mathbf{Q}_{\mathbf{g}i} = \text{diag} \underbrace{(\mathbf{Q}_i \cdots \mathbf{Q}_i)}_m \quad (114)$$

$$\bar{\mathbf{Q}}_{\mathbf{g}i} = \text{diag} \underbrace{(\mathbf{0} \mathbf{0} \cdots \bar{\mathbf{Q}}_i)}_m \quad (115)$$

$$\mathbf{R}_{\mathbf{g}i} = \text{diag} \underbrace{(\mathbf{R}_i \cdots \mathbf{R}_i)}_m \quad (116)$$

$$\Psi_{\mathbf{g}i} = \text{diag} \underbrace{(\Psi_i \cdots \Psi_i)}_m \quad (117)$$

Então:

$$\begin{aligned} V_{ki} = & \left\| \bar{\mathbf{I}}_i \left[\mathbf{x}_i^s(k) - \mathbf{y}_{ki}^{sp} - \delta_{kii} - \sum_{\substack{j=1 \\ j \neq i}}^M \delta_{kij} \right] + \Psi_{\mathbf{g}i} \mathbf{F}_{\mathbf{x}i} \mathbf{x}_i^d(k) + \right. \\ & \left. \sum_{j=1}^M \left[(\mathbf{D}_{\mathbf{m}ij}^0 + \Psi_{\mathbf{g}i} \mathbf{F}_{\mathbf{u}ij}) \Delta \mathbf{u}_{kj} \right] \right\|_{\mathbf{Q}_{\mathbf{g}i}}^2 + \left\| \mathbf{F}_{\mathbf{x}i} \mathbf{x}_i^d(k) + \sum_{j=1}^M [\mathbf{F}_{\mathbf{u}ij} \Delta \mathbf{u}_{kj}] \right\|_{\bar{\mathbf{Q}}_{\mathbf{g}i}}^2 + \\ & \left\| \Delta \mathbf{u}_{ki} \right\|_{\mathbf{R}_{\mathbf{g}i}}^2 + \sum_{j=1}^M \left\| \delta_{kij} \right\|_{\mathbf{S}_i}^2 \end{aligned} \quad (118)$$

Lembrando sempre que para o controlador do subsistema i , $\Delta \mathbf{u}_{kj}$ e δ_{kij} , $j \neq i$ são constantes a serem atualizadas oriundas do instante de amostragem $k-1$. Pode-se agora finalmente escrever a função objetivo V_k para um subsistema i qualquer:

$$\begin{aligned} V_k = & \sum_{q=1}^M w_q \left\{ \left\| \bar{\mathbf{I}}_q \left[\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \sum_{j=1}^M \delta_{kqj} \right] + \Psi_{\mathbf{g}q} \mathbf{F}_{\mathbf{x}q} \mathbf{x}_q^d(k) + \right. \right. \\ & \left. \sum_{j=1}^M \left[(\mathbf{D}_{\mathbf{m}qj}^0 + \Psi_{\mathbf{g}q} \mathbf{F}_{\mathbf{u}qj}) \Delta \mathbf{u}_{kj} \right] \right\|_{\mathbf{Q}_{\mathbf{g}q}}^2 + \left\| \mathbf{F}_{\mathbf{x}q} \mathbf{x}_q^d(k) + \sum_{j=1}^M \{\mathbf{F}_{\mathbf{u}qj} \Delta \mathbf{u}_{kj}\} \right\|_{\bar{\mathbf{Q}}_{\mathbf{g}q}}^2 + \\ & \left. \left\| \Delta \mathbf{u}_{kq} \right\|_{\mathbf{R}_{\mathbf{g}q}}^2 + \sum_{j=1}^M \left\| \delta_{kqj} \right\|_{\mathbf{S}_q}^2 \right\} \end{aligned} \quad (119)$$

Desta forma, escreve-se o problema de otimização como:

$$\min_{\Delta \mathbf{u}_{ki}, \bar{\boldsymbol{\delta}}_{ki}} V_k \quad (120)$$

$$-\Delta \mathbf{u}_i^{max} \leq \Delta \mathbf{u}_i(k+l) \leq \Delta \mathbf{u}_i^{max} \quad \forall l \quad (121)$$

$$\Delta \mathbf{u}_i(k+l) = \mathbf{0} \quad l \geq m \quad (122)$$

$$\mathbf{u}_i^{min} \leq \mathbf{u}_i(k-1) + \sum_{q=0}^l \Delta \mathbf{u}_i(k+q) \leq \mathbf{u}_i^{max} \quad l = 0, \dots, m-1 \quad (123)$$

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \boldsymbol{\delta}_{kqi} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj} - \boldsymbol{\delta}_{kqj} \right\} = \mathbf{0} \quad q = 1, \dots, M \quad (124)$$

Note que (124) nada mais é que a equação (105) reescrita explicitando-se as variáveis $\Delta \mathbf{u}_{ki}$. Nela, $\tilde{\mathbf{D}}_{ij}^0$ nada mais é que um agrupamento de $\mathbf{D}_{ij}^0$:

$$\tilde{\mathbf{D}}_{ij}^0 = \underbrace{\left[\mathbf{D}_{ij}^0 \cdots \mathbf{D}_{ij}^0 \right]}_m \quad (125)$$

O controlador repete o cálculo das equações (120) - (124) para todos os subsistemas i . Observe que, para cada subsistema i , (124) pode ser usada para zerar as variáveis de folga, de modo que $\Delta \mathbf{u}_{ki}$ é a variável de decisão do sistema.

Dada a função objetivo (119), a mesma pode ser reescrita de forma matricial. Isso, além de simplificar a notação, foi a maneira como a função objetivo foi escrita em termos de programação para as simulações computacionais que foram realizadas. O leitor que tiver interesse vide o Apêndice A.

4.1.1 Cálculo iterativo para obtenção de solução ótima

Como já dito anteriormente, o MPC relacionado ao subsistema i calcula os valores de $\Delta \mathbf{u}_{ki}$ e todas as folgas $\bar{\boldsymbol{\delta}}_{ki}$ para o seu caso em particular. À solução ótima do problema, que seria igual àquela obtida pelo sistema centralizado no instante k , dá-se o nome de $\Delta \mathbf{u}_{ki}^*$ e $\bar{\boldsymbol{\delta}}_{ki}^*$, $i = 1, \dots, M$.

No instante k , entretanto, o controlador i não dispõe dos valores de $\Delta \mathbf{u}_{kj}^*$ e $\bar{\delta}_{kj}^*$ dos outros subsistemas para serem incorporados como constantes. Estes estão disponíveis somente até o instante $k - 1$. Assim, utilizam-se valores dos subsistemas j , $j \neq i$, disponíveis do instante $k - 1$, obtendo-se uma solução diferente da solução ótima global (obtida pelo sistema centralizado). A esta solução, dá-se o nome de sub ótima. Para que a solução sub-ótima obtida convirja para a solução ótima, o cálculo feito resolvendo-se o problema (120) - (124) deve ser refeito de maneira iterativa [31].

O referido cálculo iterativo entre os subsistemas ocorre dentro de cada instante k , de modo que todos os controladores resolvem o problema (120) - (124) para seus respectivos subsistemas diversas vezes, atualizando os valores de $\Delta \mathbf{u}_{kj}$ e $\bar{\delta}_{kj}$, $j = 1, \dots, M$ a cada iteração.

Seja p a iteração atual e $\Delta \mathbf{u}_{ki}^{(p*)}$, $\bar{\delta}_{ki}^{(p*)}$ a solução obtida pelo controlador do subsistema i ao resolver o problema (120) - (124). Então, para cada controlador, a solução na iteração p , denominada $\Delta \mathbf{u}_{ki}^{(p)}$, $\bar{\delta}_{ki}^{(p)}$ será obtida por uma média entre solução $(p*)$ com os valores da iteração anterior, denominados $\Delta \mathbf{u}_{ki}^{(p-1)}$, $\bar{\delta}_{ki}^{(p-1)}$.

Seja, portanto, o seguinte algoritmo para o método iterativo:

Algoritmo 1:

1. Defina critério de parada (número fixo de iterações ou norma do erro de aproximação)
2. Estabeleça $p = 1$
3. Defina $\Delta \mathbf{u}_{ki}^{(p-1)} = [\Delta \mathbf{u}_i^T(k|k-1) \cdots \Delta \mathbf{u}_i^T(k+m-2|k-1) \mathbf{0}^T]^T$, $\bar{\delta}_{ki}^{(p-1)} = \bar{\delta}_{k-1i}$
4. Enquanto o critério de parada não for atingido, faça:
 - (a) Avalie: $(\Delta \mathbf{u}_{ki}^{(p*)}, \bar{\delta}_{ki}^{(p*)}) = \arg \min V_k(\mathbf{x}(k), \Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{kj}^{(p-1)}, j \neq i) \quad \forall i$ sujeito a (121) - (124)
 - (b) Para $i = 1, \dots, M$, estabeleça:

$$\begin{bmatrix} \Delta \mathbf{u}_{ki}^{(p)} \\ \bar{\delta}_{ki}^{(p)} \end{bmatrix} \leftarrow w_i \begin{bmatrix} \Delta \mathbf{u}_{ki}^{(p*)} \\ \bar{\delta}_{ki}^{(p*)} \end{bmatrix} + (1 - w_i) \begin{bmatrix} \Delta \mathbf{u}_{ki}^{(p-1)} \\ \bar{\delta}_{ki}^{(p-1)} \end{bmatrix}$$
 - (c) Estabeleça $p = p + 1$
5. Implemente $\Delta \mathbf{u}_i(k|k)$, $\forall i$
6. Mova para o próximo instante e retorne para 2.

Os valores de w_i são os mesmos utilizados na combinação convexa para a função objetivo. A cada iteração, a função objetivo V_k para o subsistema i recebe os novos valores das variáveis calculadas pelos outros subsistemas $j \neq i$ e os utiliza como parâmetros. Quanto ao critério de parada, o Algoritmo 1 pode ser repetido até um número predeterminado de iterações, ou então até que $\left\| \Delta \mathbf{u}_{ki}^{(p)} - \Delta \mathbf{u}_{ki}^{(p-1)} \right\|^2 \leq \epsilon$, no qual ϵ é um número pequeno. Em ambos os casos uma solução sub ótima é atingida.

4.1.2 Esquema de cascadeamento para solução ótima

Os trabalhos de Venkat *et al.* [31], Stewart *et al.* [30], Rawlings e Stewart [1] e Farina *et al.* [33] utilizam todos o arranjo no qual, entre os instantes k e $k + 1$, cada MPC obtém o ponto ótimo para a função que minimiza e na sequência troca informações com todos os outros MPCs. Esta operação de cálculo/compartilhamento de informações é iterada até que o critério de parada seja atingido. Tal arranjo adaptado para o controlador aqui definido foi discutido na subseção anterior (4.1.1).

Uma alternativa proposta por este trabalho (embora não utilizada nas simulações ao longo da dissertação) é o cálculo que pode ser chamado de “cascadeado”. Neste, dentro de uma mesma iteração p , cada MPC já envia informação para o próximo MPC assim que a obtiver. Desta forma, em p , o MPC do subsistema 1 minimiza sua função objetivo baseado nas informações da iteração anterior $p - 1$, o segundo MPC, por sua vez, dispõe das informações dos subsistemas 3, ..., M a partir do instante $p - 1$ e da informação do subsistema 1 da iteração atual p e assim por diante. O controlador M dispõe das informações de todos os subsistemas para resolver sua função objetivo. A Figura 2 ilustra comparativamente as duas sequências de cálculo.

Tal arranjo, embora impute ao sistema um maior custo computacional e uma eventual necessidade de se decidir qual sistema envia informação para qual em que ordem, pode ter a vantagem de ser mais rápido em termos de convergência, pois os $\Delta \mathbf{u}_i^{(p)*}$ calculados em cada etapa não precisam ser ponderados por w_i , já que são o ótimo daquela mesma iteração, e não da iteração anterior.

Figura 2: Comparação da arquitetura MPC tradicional vs. cascadeado

- (a) Esquema tradicional, em que as informações são trocadas por todos os MPCs após cada iteração
- (b) Esquema cascadeado, em que a informação de cada MPC é trocada dentro da mesma iteração.

Fonte: elaboração própria (2020).

4.2 Factibilidade recursiva e estabilidade do problema proposto

Definida a função objetivo para o “*Distributed and Cooperative Infinite Horizon Model Predictive Control* (DC-IHMPC)” na seção 4.1 e suas restrições, bem como o algoritmo para o método iterativo, cabe nesta seção discutir a factibilidade recursiva e a convergência do controlador, além da convergência do algoritmo proposto para o método iterativo, dando o formalismo matemático que faltou na seção anterior. Foi discutida também a convexidade do problema, uma vez que esta é importante para falar sobre convergência.

4.2.1 Convexidade do conjunto $\mathcal{U}^{nu}$ e das funções objetivo V_k

As restrições dadas por (121) - (124) se apresentam, para a variável $\Delta \mathbf{u}_{ki}$ de cada subsistema, na forma de inequações ou equações lineares. As inequações são da forma $Ax \leq b$ (x sendo a variável) e denotam um conjunto convexo que passa a ser chamado nesta seção genericamente de $\mathcal{U}_{in}^{nu_i}$. As equações, por sua vez, são da forma $Ax = b$, gerando

espaços lineares chamados genericamente $\mathcal{U}_{eq}^{nu_i}$. A intersecção do conjunto $\mathcal{U}_{in}^{nu_i}$ com o conjunto $\mathcal{U}_{eq}^{nu_i}$ gera um conjunto que é genericamente chamado de $\mathcal{U}^{nu_i}$. Este é, portanto, um conjunto convexo.

Assim, o conjunto $\mathcal{U}^{nu}$, gerado pela combinação de todos os subsistemas i , é dado por: $\mathcal{U}^{nu} = \mathcal{U}^{nu_1} \times \mathcal{U}^{nu_2} \times \dots \times \mathcal{U}^{nu_M}$, sendo portanto, também convexo. Cabe também dizer que tais conjuntos $\mathcal{U}^{nu_i}$ são não-vazios, pois $\Delta \mathbf{u}_i = \mathbf{0} \in \mathcal{U}^{nu_i} \forall i$ e as variáveis de folga são irrestritas, garantindo a factibilidade de (124). Uma vez que $\Delta \mathbf{u}_i = \mathbf{0}$ denota a ação de não alterar a entrada $\mathbf{u}_i$ naquele instante, $\Delta \mathbf{u}_i$ é sempre uma solução admissível para qualquer variável e deve pertencer a $\mathcal{U}^{nu_i}$.

Quanto às variáveis de folga, estas são irrestritas, de modo que podem assumir qualquer valor dentre os números reais. Como o conjunto dos números reais é um conjunto convexo, a presença de folgas não altera a convexidade do problema.

Com relação à função objetivo V_k , definida genericamente nas equações (82a) e (82b), ou de maneira específica em (119), esta é uma soma ponderada das funções V_{ki} , já que w_i é estritamente positivo $\forall i$ e, da sua definição, $\sum_{i=1}^M w_i = 1$. Já as funções V_{ki} , por sua vez, são sempre definidas de forma quadrática, de modo a serem convexas.

Como a combinação convexa de funções convexas também é uma função convexa [50], conclui-se que V_k é uma função convexa sobre um conjunto convexo $\mathcal{U}^{nu}$, de modo que o mínimo desta função sempre existe e é global. Os valores que minimizam a função V_k , como mencionado na subseção 4.1.1, são denominados $\Delta \mathbf{u}_{ki}^*$ e $\bar{\delta}_{ki}^*$, $i = 1, \dots, M$. A factibilidade e estabilidade do problema será demonstrada nas subseções subsequentes.

4.2.2 Convergência do modelo iterativo

Nesta subseção, mostra-se que o Algoritmo 1 descrito na subseção 4.1.1 faz o problema dado pelas equações (120) - (124) convergir para o mínimo global de V_k , o que garante que a solução ótima pode ser atingida a cada instante k . Isso é feito através dos seguintes Teoremas:

Teorema 1 - Dada a função quadrática V_k , função de todos os valores $\Delta \mathbf{u}_{kj}$, $\bar{\delta}_{kj}$, $j = 1, \dots, M$ e do seu estado inicial $x(k)$. Sejam $\Delta \mathbf{u}_{kj}^{(p)}$, $\bar{\delta}_{kj}^{(p)}$, $j = 1, \dots, M$ e $\Delta \mathbf{u}_{kj}^{(p-1)}$, $\bar{\delta}_{kj}^{(p-1)}$, $j = 1, \dots, M$ as soluções obtidas nas iterações p e $p - 1$ respectivamente. Então:

$$\begin{aligned}
& V_k \left(\Delta \mathbf{u}_{kj}^{(p)}, \bar{\delta}_{kj}^{(p)}, j = 1, \dots, M, x(k) \right) \\
& \leq V_k \left(\Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M, x(k) \right)
\end{aligned} \tag{126}$$

Prova: dados $\Delta \mathbf{u}_{ki}^{(p*)}$, $\bar{\delta}_{ki}^{(p*)}$, solução do problema V_k calculada na iteração p pelo controlador do subsistema i . Então $(x(k))$ foi omitido):

$$\begin{aligned}
& V_k \left(\Delta \mathbf{u}_{ki}^{(p*)}, \Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{ki}^{(p*)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M, j \neq i \right) \\
& \leq V_k \left(\Delta \mathbf{u}_{ki}^{(p-1)}, \Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{ki}^{(p-1)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M, j \neq i \right)
\end{aligned} \tag{127}$$

Tal equação é verdadeira pois, do Algoritmo 1 passo 4.(a): $\left(\Delta \mathbf{u}_{ki}^{(p*)}, \bar{\delta}_{ki}^{(p*)} \right) = \arg \min V_k \left(\mathbf{x}(k), \Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{kj}^{(p-1)}, j \neq i \right)$ e V_k é uma função convexa, de modo que o mínimo é global. Também do Algoritmo 1, passo 4.(b), escreve-se:

$$\begin{aligned}
& V_k \left(\Delta \mathbf{u}_{kj}^{(p)}, \bar{\delta}_{kj}^{(p)}, j = 1, \dots, M \right) \\
& = V_k \left(w_j \Delta \mathbf{u}_{kj}^{(p*)} + (1 - w_j) \Delta \mathbf{u}_{kj}^{(p-1)}, w_j \bar{\delta}_{kj}^{(p*)} + (1 - w_j) \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M \right)
\end{aligned} \tag{128}$$

Como $\sum_{q=1}^M w_q = 1 \Leftrightarrow \sum_{\substack{q=1 \\ q \neq j}}^M w_q = (1 - w_j)$, a equação (128) pode ser reescrita da seguinte forma:

$$\begin{aligned}
& V_k \left(\Delta \mathbf{u}_{kj}^{(p)}, \bar{\delta}_{kj}^{(p)}, j = 1, \dots, M \right) \\
& = V_k \left(w_j \Delta \mathbf{u}_{kj}^{(p*)} + \sum_{\substack{q=1 \\ q \neq j}}^M w_q \Delta \mathbf{u}_{kj}^{(p-1)}, w_j \bar{\delta}_{kj}^{(p*)} + \sum_{\substack{q=1 \\ q \neq j}}^M w_q \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M \right)
\end{aligned} \tag{129}$$

Lembre da subseção 2.5.1 que uma função é convexa se, e somente se, para qualquer x_1 e x_2 : $f(\lambda x_1 + (1 - \lambda) x_2) \leq \lambda f(x_1) + (1 - \lambda) f(x_2) \quad \forall \lambda \in [0, 1]$. Do fato de V_k ser convexa, o lado direito da equação (129), por sua vez, obedece à seguinte desigualdade:

$$\begin{aligned}
& V_k \left(w_j \Delta \mathbf{u}_{kj}^{(p*)} + \sum_{\substack{q=1 \\ q \neq j}}^M w_q \Delta \mathbf{u}_{kj}^{(p-1)}, w_j \bar{\delta}_{kj}^{(p*)} + \sum_{\substack{q=1 \\ q \neq j}}^M w_q \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M \right) \\
& \leq \sum_{i=1}^M w_i V_k \left(\Delta \mathbf{u}_{ki}^{(p*)}, \Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{ki}^{(p*)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M, j \neq i \right)
\end{aligned} \tag{130}$$

Pela equação (127), o lado direito da desigualdade em (130) obedece, por sua vez, à seguinte desigualdade:

$$\begin{aligned}
& \sum_{i=1}^M w_i V_k \left(\Delta \mathbf{u}_{ki}^{(p*)}, \Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{ki}^{(p*)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M, j \neq i \right) \\
& \leq \sum_{i=1}^M w_i V_k \left(\Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M \right)
\end{aligned} \tag{131}$$

Como $\sum_{i=1}^M w_i = 1$, pode-se escrever:

$$\begin{aligned}
& \sum_{i=1}^M w_i V_k \left(\Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M \right) \\
& = V_k \left(\Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M \right)
\end{aligned} \tag{132}$$

Olhando para o lado esquerdo de (129) e o lado direito de (132), e observando a sequência de igualdades e desigualdades acima, fica provada (126). Ou seja, a função objetivo V_k não cresce a cada iteração. $\square$

Dessa forma, dado o Teorema 1, pode-se definir o seguinte Teorema:

Teorema 2 - Sejam V_k uma função quadrática e $\mathcal{U}^{nu}$ um conjunto convexo e compacto. Seja $\Delta \mathbf{u}_{kj}^{(p)}, \bar{\delta}_{kj}^{(p)}, j = 1, \dots, M$ a solução obtida pelo Algoritmo 1 após p iterações. Seja também $V_k \left(\Delta \mathbf{u}_{kj}^{(p)}, \bar{\delta}_{kj}^{(p)}, j = 1, \dots, M \right)$ o custo associado a esta solução ($x(k)$ foi omitido). Sejam finalmente $\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*, j = 1, \dots, M$ e $V_k \left(\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*, j = 1, \dots, M \right)$ a solução ótima e o custo associado para o problema de minimização de V_k . Então:

$$\lim_{p \rightarrow \infty} \left(\Delta \mathbf{u}_{kj}^{(p)}, \bar{\delta}_{kj}^{(p)}, j = 1, \dots, M \right) = \left(\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*, j = 1, \dots, M \right) \quad (133)$$

$$\lim_{p \rightarrow \infty} V_k \left(\Delta \mathbf{u}_{kj}^{(p)}, \bar{\delta}_{kj}^{(p)}, j = 1, \dots, M \right) = V_k \left(\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*, j = 1, \dots, M \right) \quad (134)$$

Prova: como o conjunto de soluções admissíveis $\mathcal{U}^{nu}$ é contínuo, compacto, fechado e convexo e V_k é uma função quadrática contínua, um limite para o Algoritmo 1 existe. Sabe-se também que o problema de minimização da função V_k para o MPC tem solução única (afinal é uma função convexa sobre um conjunto convexo), denominada $\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*, j = 1, \dots, M$ e $V_k^* = V_k \left(\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*, j = 1, \dots, M \right)$.

Assuma agora que o Algoritmo 1, para p muito grande, chegue a uma solução não ótima dentro de um subconjunto de $\mathcal{U}^{nu}$ chamado $\mathcal{U}_\infty^{nu}$ denominada $V_k^\infty = V_k \left(\Delta \mathbf{u}_{kj}^\infty, \bar{\delta}_{kj}^\infty, j = 1, \dots, M \right)$. Pela convexidade do problema: $V_k^\infty \geq V_k^*$. Para estabelecer convergência, assume-se que o problema não converge e prova-se por uma contradição.

Se, por hipótese, o problema não convergir, isso implica que existe uma solução diferente gerada por uma nova iteração denominada $\Delta \mathbf{v}_{kj}, \bar{\lambda}_{kj}, j = 1, \dots, M$. Seja esta nova solução dada pelo Algoritmo 1 passo 4.(b):

$$\begin{bmatrix} \Delta \mathbf{v}_{kj} \\ \bar{\lambda}_{kj} \end{bmatrix} \leftarrow w_j \begin{bmatrix} \Delta \mathbf{z}_{kj} \\ \bar{\kappa}_{kj} \end{bmatrix} + (1 - w_j) \begin{bmatrix} \Delta \mathbf{u}_{kj}^\infty \\ \bar{\delta}_{kj}^\infty \end{bmatrix} \quad j = 1, \dots, M \quad (135)$$

Em que $\Delta \mathbf{z}_{kj}, \bar{\kappa}_{kj}, j = 1, \dots, M$ é a solução obtida por cada um dos controladores na presente iteração resolvendo (120)-(124), Algoritmo 1 passo 4.(a):

$$(\Delta \mathbf{z}_{kj}, \bar{\kappa}_{kj}) = \arg \min V_k \left(\Delta \mathbf{u}_{kq}^\infty, \bar{\delta}_{kq}^\infty, q \neq j \right) \quad j = 1, \dots, M \quad (136)$$

Sujeito às restrições (121) - (124). Como, por hipótese, o problema não converge, há pelo menos um subsistema j no qual:

$$\begin{bmatrix} \Delta \mathbf{z}_{kj} \\ \bar{\kappa}_{kj} \end{bmatrix} \neq \begin{bmatrix} \Delta \mathbf{u}_{kj}^\infty \\ \bar{\delta}_{kj}^\infty \end{bmatrix} \quad (137)$$

Observe também que as três soluções $(\Delta \mathbf{v}_{kj}, \bar{\lambda}_{kj})$, $(\Delta \mathbf{z}_{kj}, \bar{\kappa}_{kj})$ e $(\Delta \mathbf{u}_{kj}, \bar{\delta}_{kj})$, $j = 1, \dots, M$ pertencem a $\mathcal{U}_\infty^{nu}$, pela convexidade do problema e pelo fato de $\mathcal{U}_\infty^{nu}$ poder se tornar quão pequeno quanto se quiser bastando para isso aumentar que o número de iterações seja

grande. Assim: $V_k(\Delta \mathbf{v}_{kj}, \bar{\boldsymbol{\lambda}}_{kj}, j = 1, \dots, M) = V_k^\infty$. Deste modo:

$$\begin{aligned}
V_k^\infty &= V_k(\Delta \mathbf{v}_{kj}, \bar{\boldsymbol{\lambda}}_{kj}, j = 1, \dots, M) \\
&= V_k(w_j \Delta \mathbf{z}_{kj} + (1 - w_j) \Delta \mathbf{u}_{kj}^\infty, w_j \bar{\boldsymbol{\kappa}}_{kj} + (1 - w_j) \bar{\boldsymbol{\delta}}_{kj}^\infty, j = 1, \dots, M) \\
&= V_k \left(w_j \Delta \mathbf{z}_{kj} + \sum_{\substack{q=1 \\ q \neq j}}^M w_q \Delta \mathbf{u}_{kj}^\infty, w_j \bar{\boldsymbol{\kappa}}_{kj} + \sum_{\substack{q=1 \\ q \neq j}}^M w_q \bar{\boldsymbol{\delta}}_{kj}^\infty, j = 1, \dots, M \right) \\
&\leq \sum_{q=1}^M w_q V_k(\Delta \bar{\mathbf{z}}_{kq}, \Delta \mathbf{u}_{kj}^\infty, \bar{\boldsymbol{\kappa}}_{kq}, \bar{\boldsymbol{\delta}}_{kj}^\infty, j \neq q) \\
&\leq \sum_{q=1}^M w_q V_k^\infty = V_k^\infty
\end{aligned} \tag{138}$$

A última igualdade obedece à mesma passagem de (128) para (129). A primeira desigualdade é um passo análogo ao da passagem de (129) para (130). A segunda desigualdade vem da hipótese (137) para pelo menos um valor de $j = 1, \dots, M$. Assim, chega-se a uma contradição, de modo que o Algoritmo 1 converge.

Suponha agora que este limite para o qual o Algoritmo 1 converge é tal que $(\Delta \mathbf{u}_{kj}^\infty, \bar{\boldsymbol{\delta}}_{kj}^\infty) \neq (\Delta \mathbf{u}_{kj}^*, \bar{\boldsymbol{\delta}}_{kj}^*)$, para pelo menos um valor j . Então, pela convexidade de V_k : $V_k(\Delta \mathbf{u}_{kj}^\infty, \bar{\boldsymbol{\delta}}_{kj}^\infty, j = 1, \dots, M) = V_k^\infty \geq V_k^*$. Também pela hipótese acima, $\Delta \mathbf{u}_{kj}^* - \Delta \mathbf{u}_{kj}^\infty \neq \mathbf{0}$ e $\bar{\boldsymbol{\delta}}_{kj}^* - \bar{\boldsymbol{\delta}}_{kj}^\infty \neq \mathbf{0}$ para pelo menos um valor $j = 1, \dots, M$. Tais diferenças são definidas como: $\Delta \mathbf{u}_{kj}^* - \Delta \mathbf{u}_{kj}^\infty = \bar{\Delta} \mathbf{u}_{kj}$ e $\bar{\boldsymbol{\delta}}_{kj}^* - \bar{\boldsymbol{\delta}}_{kj}^\infty = \bar{\Delta} \bar{\boldsymbol{\delta}}_{kj}$.

Uma expansão de segunda ordem da série de Taylor ao redor de $(\Delta \mathbf{u}_{kj}^\infty, \bar{\boldsymbol{\delta}}_{kj}^\infty)$, $j = 1, \dots, M$, em que $\mathbf{H}(V_k)$ é o Hessiano da função objetivo, resulta em:

$$\begin{aligned}
&V_k(\Delta \mathbf{u}_{kj}^*, \bar{\boldsymbol{\delta}}_{kj}^*, j = 1, \dots, M) \\
&= V_k(\Delta \mathbf{u}_{kj}^\infty + \bar{\Delta} \mathbf{u}_{kj}, \bar{\boldsymbol{\delta}}_{kj}^\infty + \bar{\Delta} \bar{\boldsymbol{\delta}}_{kj}, j = 1, \dots, M) \\
&= V_k(\Delta \mathbf{u}_{kj}^\infty, \bar{\boldsymbol{\delta}}_{kj}^\infty, j = 1, \dots, M) + \\
&\quad \sum_{q=1}^M \nabla_{\Delta \mathbf{u}_{kq}} V_k(\Delta \mathbf{u}_{kj}^\infty, \bar{\boldsymbol{\delta}}_{kj}^\infty, j = 1, \dots, M) \bar{\Delta} \mathbf{u}_{kq} + \\
&\quad \sum_{q=1}^M \nabla_{\bar{\boldsymbol{\delta}}_{kj}} V_k(\Delta \mathbf{u}_{kj}^\infty, \bar{\boldsymbol{\delta}}_{kj}^\infty, j = 1, \dots, M) \bar{\Delta} \bar{\boldsymbol{\delta}}_{kq} + \\
&\quad \frac{1}{2} \left\| \begin{bmatrix} \bar{\Delta} \mathbf{u}_{k1}^T & \dots & \bar{\Delta} \mathbf{u}_{kM}^T & \bar{\Delta} \bar{\boldsymbol{\delta}}_{k1}^T & \dots & \bar{\Delta} \bar{\boldsymbol{\delta}}_{kM}^T \end{bmatrix}^T \right\|_{\mathbf{H}(V_k)}^2
\end{aligned} \tag{139}$$

Como $(\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*)$, $j = 1, \dots, M$ é a solução ótima que minimiza V_k , os termos linear (os dois somatórios do lado direito) e quadrático (último termo do lado direito) de (139) devem ser negativos ou nulos. Entretanto, para qualquer função f em um conjunto convexo X , $\nabla f(x^*)(x - x^*) \geq 0$, $\forall x \in X$ [58], de modo que o termo linear de (139) é não-negativo. O termo quadrático de (139) também é não-negativo porque V_k é quadrática (Matriz Hessiana positiva semidefinida). Dessa forma, tais termos são nulos de modo que $V_k(\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*, j = 1, \dots, M) = V_k(\Delta \mathbf{u}_{kj}^\infty, \bar{\delta}_{kj}^\infty, j = 1, \dots, M)$ e portanto, $(\Delta \mathbf{u}_{kj}^*, \bar{\delta}_{kj}^*) = (\Delta \mathbf{u}_{kj}^\infty, \bar{\delta}_{kj}^\infty)$, $j = 1, \dots, M$. Dessa forma, (133) e (134) ficam provadas. $\square$

4.2.3 Convergência do modelo iterativo no arranjo cascadeado

Em termos de convergência do modelo iterativo, basta agora falar sobre a convergência do arranjo cascadeado. Isso também é feito através do Teorema a seguir.

Teorema 3 - Para o arranjo definido na subseção 4.1.2, o custo da função V_k se comporta de tal maneira de uma iteração para a outra, que também são válidas as equações (126), (133) e (134) para este arranjo.

Prova: observando-se a equação (127), percebe-se que, se o custo global calculado pelo sistema i na iteração p é menor do que o custo global obtido na iteração anterior $p - 1$, então naturalmente, caso outros controladores resolvam cada um na sequência a mesma função V_k manipulando diferentes conjuntos de variáveis, o custo também diminui. Para ilustrar tal situação, inicia-se a otimização pelo subsistema número 1, passando em seguida para o número 2 e assim por diante até chegar a M , como ilustrado na equação (140):

$$\begin{aligned}
& V_k \left(\Delta \mathbf{u}_{k1}^{(p-1)}, \dots, \Delta \mathbf{u}_{kM}^{(p-1)}, \bar{\delta}_{k1}^{(p-1)}, \dots, \bar{\delta}_{kM}^{(p-1)} \right) \\
& \geq V_k \left(\Delta \mathbf{u}_{k1}^{(p*)}, \Delta \mathbf{u}_{k2}^{(p-1)}, \dots, \Delta \mathbf{u}_{kM}^{(p-1)}, \bar{\delta}_{k1}^{(p*)}, \bar{\delta}_{k2}^{(p-1)}, \dots, \bar{\delta}_{kM}^{(p-1)} \right) \\
& \geq V_k \left(\Delta \mathbf{u}_{k1}^{(p*)}, \Delta \mathbf{u}_{k2}^{(p*)}, \Delta \mathbf{u}_{k3}^{(p-1)}, \dots, \Delta \mathbf{u}_{kM}^{(p-1)}, \bar{\delta}_{k1}^{(p*)}, \bar{\delta}_{k2}^{(p*)}, \bar{\delta}_{k3}^{(p-1)}, \dots, \bar{\delta}_{kM}^{(p-1)} \right) \\
& \geq \dots \\
& \geq V_k \left(\Delta \mathbf{u}_{k1}^{(p*)}, \dots, \Delta \mathbf{u}_{kM}^{(p*)}, \bar{\delta}_{k1}^{(p*)}, \dots, \bar{\delta}_{kM}^{(p*)} \right)
\end{aligned} \tag{140}$$

A partir daí, pode-se utilizar o Algoritmo 1 alterando o passo 4.(a) para o arranjo cas-

cateado obtendo-se uma sequência similar à utilizada nas equações (127) - (132). Nesse caso, porém, (127) é substituída por (140), (128) - (130) se mantêm inalteradas e (131) passa a ser:

$$\begin{aligned} & \sum_{i=1}^M w_i V_k \left(\Delta \mathbf{u}_{kj}^{(p*)}, \bar{\delta}_{kj}^{(p*)}, j = 1, \dots, M \right) \\ & \leq \sum_{i=1}^M w_i V_k \left(\Delta \mathbf{u}_{kj}^{(p-1)}, \bar{\delta}_{kj}^{(p-1)}, j = 1, \dots, M \right) \end{aligned} \quad (141)$$

Ou seja, vale também (132). Finalmente, as equações (133) e (134) referentes ao Teorema 2 também valem, pois a função V_k é a mesma e continua convexa, o conjunto $\mathcal{U}^{nu}$ também não se altera, permanecendo convexo, compacto e fechado e o custo continua a decair de uma iteração para a outra, de modo que a prova é idêntica à do Teorema 2. Ou seja, tal arranjo é uma alternativa àquela utilizada por Venkat *et al.* [31]. $\square$

4.2.4 A questão das restrições e o modelo iterativo

Restrições na forma de inequações como as utilizadas neste trabalho também são utilizadas em outros controladores, como é o caso de Stewart *et al.* [30] e Venkat *et al.* [31]. O Algoritmo 1 não corre o risco de violar as restrições (121) e (123). Para ilustrar isso, seja a restrição (121) e a solução do controlador em uma determinada iteração p tal que:

$$-\Delta \mathbf{u}_i^{max} \leq \Delta \mathbf{u}_i^{(p*)} \leq \Delta \mathbf{u}_i^{max} \quad (142)$$

Multiplique esta inequação agora por w_i . Sendo este um número estritamente positivo, a inequação é mantida:

$$-w_i \Delta \mathbf{u}_i^{max} \leq w_i \Delta \mathbf{u}_i^{(p*)} \leq w_i \Delta \mathbf{u}_i^{max} \quad (143)$$

Suponha também que na iteração $p - 1$ o controlador conseguiu manter a inequação respeitada, tal que:

$$-\Delta \mathbf{u}_i^{max} \leq \Delta \mathbf{u}_i^{(p-1)} \leq \Delta \mathbf{u}_i^{max} \quad (144)$$

Multiplique agora esta inequação por $(1 - w_i)$. Este também é estritamente positivo, assim:

$$-(1 - w_i)\Delta \mathbf{u}_i^{max} \leq (1 - w_i)\Delta \mathbf{u}_i^{(p-1)} \leq (1 - w_i)\Delta \mathbf{u}_i^{max} \quad (145)$$

Somando agora (143) e (145), obtém-se a seguinte inequação:

$$-\Delta \mathbf{u}_i^{max} \leq w_i \Delta \mathbf{u}_i^{(p*)} + (1 - w_i)\Delta \mathbf{u}_i^{(p-1)} \leq \Delta \mathbf{u}_i^{max} \Rightarrow -\Delta \mathbf{u}_i^{max} \leq \Delta \mathbf{u}_i^{(p)} \leq \Delta \mathbf{u}_i^{max} \quad (146)$$

Ou seja, bastando que na primeira iteração $p = 1$ o valor inicial obedeça a (144), todos os valores para as outras iterações também devem obedecer. Como o valor inicial escolhido é aquele determinado no Algoritmo 1 passo 3, que é a solução do instante anterior deslocada no tempo, a restrição (121) está garantida. O raciocínio é análogo para a restrição (123). Já a restrição (122) é inerente à programação do controlador.

Quanto à (124), diferentemente de outros controladores distribuídos, o controlador aqui proposto contém variáveis de folga com o intuito de garantir que o problema seja factível. Mais ainda, o uso destas variáveis associado ao fato de o controlador ter um horizonte infinito de predição requer que sejam utilizadas restrições na forma de igualdades. Para cada controlador, M variáveis de folga e consequentemente M restrições de igualdade são aplicadas, totalizando-se M^2 variáveis e M^2 equações para todo o sistema.

A cada iteração, o controlador do subsistema i resolve o problema dado por (120) - (124). Assuma que (124) é válida em $p - 1$:

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \boldsymbol{\delta}_{kqi}^{(p-1)} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki}^{(p-1)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kqj}^{(p-1)} \right\} = \mathbf{0} \quad q = 1, \dots, M \quad (147)$$

Seja agora a solução calculada na iteração p para o subsistema i (Algoritmo 1 passo 4.(a)) tal que:

$$\mathbf{x}_i^s(k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii}^{(p*)} + \tilde{\mathbf{D}}_{ii}^0 \Delta \mathbf{u}_{ki}^{(p*)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{ij}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kij}^{(p-1)} \right\} = \mathbf{0} \quad (148a)$$

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \boldsymbol{\delta}_{kqi}^{(p*)} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki}^{(p*)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kqj}^{(p-1)} \right\} = \mathbf{0} \quad q \neq i \quad (148b)$$

Então, multiplique (147) por $(1 - w_q)$, (148a) por w_i e (148b) por w_q e some cada equação em (147) com as equações (148a) e (148b) obedecendo os índices corretamente. Dessa forma, obtêm-se:

$$\mathbf{x}_i^s(k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii}^{(p)} + \tilde{\mathbf{D}}_{ii}^0 \Delta \mathbf{u}_{ki}^{(p)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{ij}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kij}^{(p-1)} \right\} = \mathbf{0} \quad (149a)$$

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \boldsymbol{\delta}_{kqi}^{(p)} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki}^{(p)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kqj}^{(p-1)} \right\} = \mathbf{0} \quad q \neq i \quad (149b)$$

Note que (149a) e (149b) são as M equações para o subsistema i , mas o mesmo é válido para todos os outros subsistemas. Assim, repetindo esta mesma conta (147) - (149b) para os outros subsistemas e depois somando cada equação (149a) e (149b) de cada subsistema com a equação de mesmo índice para os outros subsistemas, chega-se a:

$$\begin{aligned} & M \left[\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} \right] - \left[(M-1) \boldsymbol{\delta}_{kqi}^{(p-1)} + \boldsymbol{\delta}_{kqi}^{(p)} \right] + \\ & \tilde{\mathbf{D}}_{qi}^0 \left[(M-1) \Delta \mathbf{u}_{ki}^{(p-1)} + \Delta \mathbf{u}_{ki}^{(p)} \right] + \\ & \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \left[(M-1) \Delta \mathbf{u}_{kj}^{(p-1)} + \Delta \mathbf{u}_{kj}^{(p)} \right] - \right. \\ & \left. \left[(M-1) \boldsymbol{\delta}_{kqj}^{(p-1)} + \boldsymbol{\delta}_{kqj}^{(p)} \right] \right\} = \mathbf{0} \end{aligned} \quad q = 1, \dots, M \quad (150)$$

Finalmente, subtraindo (147) de (150) $(M-1)$ vezes, obtêm-se:

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \boldsymbol{\delta}_{kqi}^{(p)} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki}^{(p)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj}^{(p)} - \boldsymbol{\delta}_{kqj}^{(p)} \right\} = \mathbf{0} \quad q = 1, \dots, M \quad (151)$$

que é (147) em p .

4.2.5 Factibilidade recursiva do controlador proposto

Para abordar factibilidade recursiva, assumiu-se que, para um número suficientemente grande de iterações, o Algoritmo 1 converge para a solução ótima (como demonstrado nos Teoremas 1 e 2). A prova entretanto, é válida mesmo para o caso em que uma solução sub ótima seja obtida. Como pelos Teoremas 1 e 2, soluções ótimas ou pelo menos sub ótimas são sempre o caso, a factibilidade recursiva é garantida.

Para o teorema e as provas a seguir, tenha em mente que, como discutido na subseção 4.2.1, o problema de minimização da função V_k é convexo sobre um conjunto convexo e tem solução única.

Teorema 4 - Para um sistema estável, não perturbado, com modelo nominal e estado medido, se o controlador definido através das equações (120) - (124) e do Algoritmo 1 é factível no instante k , ele permanecerá factível nos instantes subsequentes.

Prova: utilizando-se da mesma notação das seções anteriores, suponha então que $\Delta \mathbf{u}_{ki}^*$, $\bar{\boldsymbol{\delta}}_{ki}^*$, seja a solução ótima (e portanto global) para o problema de o controlador i minimizar V_k no instante de tempo k (obtida por um número suficientemente grande de iterações p). Assim, o movimento $\Delta \mathbf{u}_i^*(k|k)$ é injetado no sistema no instante k para todos os subsistemas $i = 1, \dots, M$.

Considere a seguir a sequência $\Delta \tilde{\mathbf{u}}_{k+1i} = [\Delta \mathbf{u}_i^{*T}(k+1|k) \Delta \mathbf{u}_i^{*T}(k+2|k) \dots \Delta \mathbf{u}_i^{*T}(k+m-1|k) \mathbf{0}^T]^T$, $\tilde{\boldsymbol{\delta}}_{k+1i} = \bar{\boldsymbol{\delta}}_{ki}^*$, $i = 1, \dots, M$. É trivial mostrar que tal sequência obedece às restrições (121) - (123), já que $\mathbf{0} \in \mathcal{U}^{nu}$, como discutido anteriormente e as variáveis de folga são irrestritas.

Para mostrar que (124) também é satisfeita, considere a sequência acima e a seguinte expressão algébrica:

$$\mathbf{x}_q^s(k+1) - \mathbf{y}_{kq}^{sp} - \tilde{\boldsymbol{\delta}}_{k+1qi} + \tilde{\mathbf{D}}_{qi}^0 \Delta \tilde{\mathbf{u}}_{k+1i} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \tilde{\mathbf{u}}_{k+1j} - \tilde{\boldsymbol{\delta}}_{k+1qj} \right\} \quad q = 1, \dots, M \quad (152)$$

Observe que: $\mathbf{x}_i^s(k+1) = \mathbf{x}_i^s(k) + \sum_{j=1}^M \mathbf{D}_{ij}^0 \Delta \mathbf{u}_j^*(k|k)$, $i = 1, \dots, M$. Substituindo este resultado em (152), obtém-se a mesma expressão do lado esquerdo de (124), que é nula. Logo, $\Delta \tilde{\mathbf{u}}_{k+1i}$, $\tilde{\boldsymbol{\delta}}_{k+1i}$ é uma solução factível. Havendo pelo menos uma solução factível para $k+1$, pode-se fazer o mesmo raciocínio recursivamente para $k+2$ e assim por diante, de modo que a factibilidade recursiva está provada. $\square$

4.2.6 Estabilidade do controlador proposto

Nas seções anteriores, discutiu-se a convexidade da função V_k e do conjunto $\mathcal{U}^{nu}$, o que permitiu mostrar que o método iterativo proposto converge, respeita as restrições e pode levar à solução ótima do problema a cada instante k , que no caso do MPC, é única. Nesta seção, por sua vez, pretendeu-se mostrar que a função objetivo diminui de cada instante k para $k+1$. Com isso, prova-se que o controlador é estável e que é capaz de levar uma determinada planta a qualquer set-point desejado, desde que o mesmo seja factível. Mais uma vez, isso vai ser feito na forma de Teorema.

Teorema 5 - Para um sistema estável, modelo nominal, não perturbado, se o estado estacionário desejado é atingível, o controlador produzido pela solução do problema definido pelas equações (120) - (124) (com matrizes de variáveis de folga $\mathbf{S}_i$ positivas e definidas) e pelo Algoritmo 1 é assintoticamente estável e as saídas do sistema são levadas para o set-point desejado. Se o set-point desejado não for atingível, o controlador estabiliza o sistema em um estado estacionário que minimiza a distância do mesmo para o estado desejado.

Prova: dado $V_k^* = V_k(\Delta \mathbf{u}_{ki}^*, \bar{\boldsymbol{\delta}}_{ki}^*)$ o valor da função objetivo obtido pelo controlador i em k após um número suficientemente grande de iterações p (Algoritmo 1) considerando (120) - (124). Seja $\tilde{V}_{k+1}$ a solução obtida pela sequência $\Delta \tilde{\mathbf{u}}_{k+1i}$, $\tilde{\boldsymbol{\delta}}_{k+1i}$ definida na subseção 4.2.5, para este mesmo controlador. É fácil observar que:

$$\tilde{V}_{k+1} - V_k^* = \sum_{q=1}^M w_q \left\{ - \left\| \mathbf{y}_q(k|k) - \mathbf{y}_{kq}^{sp} - \sum_{j=1}^M \delta_{kqj}^* \right\|_{\mathbf{Q}_q}^2 \right\} - \|\Delta \mathbf{u}_i^*(k|k)\|_{\mathbf{R}_i}^2 \quad (153)$$

Como as matrizes $\mathbf{Q}_q$, $\mathbf{R}_q$ são positivas, enquanto $\mathbf{y}_q(k|k) - \mathbf{y}_{kq}^{sp} - \sum_{j=1}^M \delta_{kqj}^*$, ou $\Delta \mathbf{u}_i^*(k|k)$ não forem nulos:

$$\tilde{V}_{k+1} - V_k^* \leq 0 \Leftrightarrow \tilde{V}_{k+1} \leq V_k^* \quad (154)$$

Consequentemente:

$$V_{k+1}^* \leq V_k^* \quad (155)$$

Tal situação se repete para $k+1$, $k+2$ e assim sucessivamente. Desta forma, fica mostrado que a função objetivo é monotonicamente decrescente. $\square$

4.3 Resultados de simulação para sistemas simplificados

Todas as simulações da dissertação consideram que os modelos são nominais e utilizam Filtro de Kalman conforme descrito na seção 3.3. Como simulações iniciais, foram testados sistemas simplificados, a saber, sistemas 2×2 e 3×3 . Tais simulações tinham por objetivo avaliar o funcionamento e as características do controlador e estudá-las antes de passar para simulações de casos industriais. Pelo mesmo motivo, é interessante mostrar tais resultados na dissertação. Quanto ao método iterativo, foi utilizado o Algoritmo 1 desenvolvido na subseção 4.1.1. Para a primeira simulação, utilizou-se o sistema 2×2 simulando 150 minutos de operação:

$$\mathbf{G}(s) = \begin{bmatrix} \frac{-2.5}{9.05s + 1} & \frac{5.8}{9s + 1} \\ \frac{-1.5}{7.5s + 1} & \frac{5.3}{10.2s + 1} \end{bmatrix} \quad (156)$$

Os parâmetros foram os seguintes:

$$\Delta T = 1 \text{ min}$$

$$\mathbf{y}_0 = [0 \ 0]^T$$

$$\mathbf{y}_k^{sp} = [0.5 \ -0.1]^T$$

$$\mathbf{u}^{max} = [1 \ 1]^T$$

$$\mathbf{u}^{min} = [-1 \ -1]^T$$

$$\Delta \mathbf{u}^{max} = [0.1 \ 0.1]^T$$

$$\mathbf{Q} = \text{diag}(1 \ 1)$$

$$\mathbf{R} = \text{diag}(1 \ 1)$$

$$\mathbf{S} = \text{diag}(100 \ 100)$$

Os pares escolhidos para compor cada subsistema foram: (y_1, u_1) e (y_2, u_2) . Ou seja, o primeiro MPC manipula u_1 e o segundo MPC manipula u_2 . Denominam-se interações cruzadas aquelas relacionadas a entradas e saídas fora do mesmo subsistema, no caso (y_1, u_2) e (y_2, u_1) . Observe que as mesmas são bastante fortes, de modo que um controlador descentralizado poderia se tornar instável.

Os resultados preliminares são exibidos nas Figuras 3 a 6, em que há uma comparação entre o controlador centralizado e os casos estudados para o controlador distribuído. Variaram-se parâmetros como o número de iterações p e o peso $\mathbf{w}$. As Figuras 3 e 4 mostram os efeitos da alteração do parâmetro p . Nestas simulações, número fixo de iterações foi o único critério de parada adotado.

Figura 3: Simulação: sistema 2×2 - MPC centralizado vs. distribuído - variação de p

Fonte: elaboração própria (2020).

Como se pode observar na Figura 3, o aumento do número de iterações, por sua vez, faz com que o DC-IHMPC se aproxime de um IHMPC centralizado. Nesta simulação foram utilizados pesos iguais $\mathbf{w} = [0.5 \ 0.5]^T$, $m = 2$ e número de iterações $p = 2$, $p = 4$ e $p = 8$.

Com relação aos tempos de computação, as simulações para os controladores distribuídos foram feitas um subsistema após o outro no Matlab e o tempo total de simulação foi dividido pelo número de subsistemas para simular o tempo que levaria caso cada controlador rodasse em um servidor diferente. Foram feitas também várias simulações por sistema e o tempo reportado é uma média de cada valor obtido.

Assim, o controlador centralizado levou 12.20ms em média a cada instante para computar a solução, já o controlador distribuído levou 14.08ms em média a cada instante para $p = 2$ (já considerando um controlador por servidor), 30.10ms em média a cada instante para $p = 4$ e 60.62ms em média a cada instante para $p = 8$. Para $p = 2$, o tempo de simulação é similar ao do sistema centralizado, mas o desempenho fica bastante prejudicado. Em casos mais próximos de uma aplicação industrial, tal situação não se mantém. De fato, um sistema 2×2 seria muito simples para ser subdividido na prática.

Este sistema em particular mostrou uma convergência muito lenta quando $p = 2$. Tal resultado não é observado em todas as simulações e se deve provavelmente ao fato de as interações cruzadas (y_1, u_2) e (y_2, u_1) serem muito fortes. A mesma tendência se pode observar na função objetivo, que converge em um tempo menor de simulação conforme o número de iterações aumenta.

Figura 4: Custo: sistema 2×2 - MPC centralizado vs. distribuído - variação de p .

Fonte: elaboração própria (2020).

Com relação à mudança nos valores de $\mathbf{w}$, quando o peso das variáveis é distribuído de maneira desigual, espera-se que haja convergência mais rápida da variável com o maior peso, embora isso ocorra, o controlador com pesos iguais mostrou-se mais rápido do que

o de pesos desiguais com relação à função objetivo (Figuras 5 e 6). Neste caso utilizou-se $m = 2$, $p = 4$ e variação de pesos: $\mathbf{w} = [0.5 \ 0.5]^T$, $\mathbf{w} = [0.7 \ 0.3]^T$ e $\mathbf{w} = [0.3 \ 0.7]^T$.

Figura 5: Simulação: sistema 2×2 - MPC centralizado vs. distribuído - variação de $\mathbf{w}$

Fonte: elaboração própria (2020).

Figura 6: Custo: sistema 2×2 - MPC centralizado vs. distribuído - variação de $\mathbf{w}$

Fonte: elaboração própria (2020).

Com relação aos tempos computacionais, o controlador centralizado levou 12.20ms em média a cada instante para computar a solução, já o controlador distribuído levou 30.10ms

em média a cada instante para $\mathbf{w} = [0.5 \ 0.5]^T$ (valores obtidos da mesma forma que para o sistema 2×2), 31.89ms em média a cada instante para $\mathbf{w} = [0.7 \ 0.3]^T$ e 32.88ms em média a cada instante para $\mathbf{w} = [0.3 \ 0.7]^T$. Embora o tempo computacional seja parecido, os controladores com pesos desiguais tiveram ambos resultados levemente piores do que o controlador com pesos iguais.

De fato, variar os pesos não é a melhor forma de se obter maior prioridade em determinadas variáveis, já que a própria matriz de pesos $\mathbf{Q}$ faz isso de maneira muito mais eficiente, atribuindo peso durante a resolução do problema de otimização, e não durante o modelo iterativo.

Esta primeira simulação mostra que o controlador funciona bem, mesmo em casos sub ótimos ele consegue levar as saídas para seus set-points rapidamente (com exceção a $p = 2$). Mais ainda, ela confirma uma tendência já esperada, como aproximação do ponto ótimo para maior número de iterações.

Passando para uma situação um pouco mais complexa, em que há outras particularidades para se mostrar, seja o caso 3×3 , já com as FTs em tempo discreto dado pela equação seguinte:

$$\mathbf{G}(z) = \begin{bmatrix} \frac{0.4758}{z - 0.9048} & \frac{0.94}{z - 0.8825} & 0 \\ 0 & \frac{0.2128}{z - 0.9355} & \frac{0.6661}{z - 0.9048} \\ \frac{0.387}{z - 0.9355} & 0 & \frac{0.9754}{z - 0.9512} \end{bmatrix} \quad (157)$$

Sejam os seguintes parâmetros:

$$\begin{aligned} \Delta T &= 1\text{min} & p &= 4 \\ \mathbf{y}_0 &= [1 \ 1 \ 1]^T & m &= 3 \\ \mathbf{y}_k^{sp} &= [2 \ 2 \ 1]^T & \mathbf{Q} &= \text{diag}(1 \ 1 \ 1) \\ \mathbf{u}^{max} &= [0.5 \ 0.5 \ 0.1]^T & \mathbf{R} &= \text{diag}(1 \ 1 \ 1) \\ \mathbf{u}^{min} &= -\mathbf{u}^{max} & \mathbf{S} &= \text{diag}(100 \ 100 \ 100) \\ \Delta \mathbf{u}^{max} &= [0.05 \ 0.05 \ 0.01]^T \end{aligned}$$

Tal sistema pode ser subdividido de diversas formas. Primeiramente, pode-se permutar as combinações entrada/saída. Por exemplo, duas possíveis combinações são: (y_1, u_1) , (y_2, u_2) , (y_3, u_3) e (y_1, u_3) , (y_2, u_1) , (y_3, u_2) . Outra maneira de variar as combinações é variar o tamanho dos subsistemas. No caso 3×3 pode-se fazer um subsistema com

tamanho 2×2 e outro 1×1 . Um exemplo deste tipo de arranjo seria (y_1, y_2, u_1, u_2) , (y_3, u_3) .

Os subsistemas que têm mais de um par entrada/saída funcionam, internamente, como se fossem MPCs centralizados. Seu caráter descentralizado aparece somente em relação aos outros subsistemas. Tal situação permite uma flexibilidade grande de operação. Ela permite também organizar quais sistemas precisam ou não estar juntos e quais combinações são mais importantes.

Nas Figuras 7 e 8, podem-se observar os resultados de simulação comparativos do sistema descrito em (157) em sua forma totalmente distribuída - três sistemas 1×1 , sendo estes: (y_1, u_1) , (y_2, u_2) e (y_3, u_3) denominado $subd = 3$ - e em sua forma parcialmente distribuída - (y_1, y_2, u_1, u_2) e (y_3, u_3) denominado $subd = 2$ - além, é claro, da comparação com o sistema centralizado. Os pesos para $subd = 2$ foram $\mathbf{w} = [0.5 \ 0.5]^T$ e para $subd = 3$ foram $\mathbf{w} = [1/3 \ 1/3 \ 1/3]^T$.

Figura 7: Simulação: sistema 3×3 - MPC centralizado vs. distribuído

Fonte: elaboração própria (2020).

Figura 8: Custos: sistema 3×3 - MPC centralizado vs. distribuído

Fonte: elaboração própria (2020).

Como era de se esperar, o sistema dividido em dois subsistemas se aproximou mais do sistema centralizado do que o sistema dividido em três subsistemas. Como foi dito anteriormente, o número de subsistemas funciona aqui como um grau de descentralização. Variáveis com interações muito fortes podem ser agrupadas em um mesmo subsistema, de modo que internamente a solução ótima entre elas seja obtida sempre, mesmo com baixo número de iterações. Com isso, mostra-se o grau de flexibilidade de um sistema distribuído.

Com relação aos tempos computacionais, o controlador centralizado levou 7.12ms em média a cada instante para computar a solução, já o controlador distribuído levou 16.94ms em média a cada instante para $subd = 2$ e 10.22ms em média a cada instante para $subd = 3$. Como esperado, quanto maior o número de subdivisões, menor o tempo de computação, pois a estrutura de controle é dividida em mais servidores.

Note que, mesmo com $p = 4$, o controlador subdividido em três subsistemas teve desempenho um pouco inferior ao controlador subdividido em dois subsistemas. Mais uma vez, estes casos são bastante simples, o que faz com que os controladores distribuídos tenham desempenho inferior ao centralizado com relação ao tempo computacional.

5 Proposta para o controlador DC-IHMPC com RTO

Poucos trabalhos sobre MPC distribuído focam no controlador com camadas de otimização superiores na estrutura de controle. Há somente algumas tentativas de realizar tal utilização [59, 60]. A alternativa aqui proposta, baseada no trabalho de González e Odloak [25], teve por objetivo preencher esta lacuna.

Dito isto, discutiu-se neste capítulo a criação de um controlador distribuído e cooperativo próprio para operar com o RTO. Tal controlador utiliza o modelo definido na seção 3.1, apresenta faixas nas saídas e alvos (em inglês: “targets”) nas entradas, podendo operar com menor número de entradas do que de saídas. Ele também se diferencia daquele apresentado no Capítulo 4, que necessita operar em um sistema quadrado e não permite integração com camadas de otimização.

Assim como no capítulo anterior, este capítulo tem a formulação do controlador apresentada na primeira parte (seção 5.1) bem como o algoritmo para o método iterativo do controlador distribuído (subseção 5.1.1). A seguir, são abordadas questões como factibilidade recursiva e estabilidade (seção 5.2). Finalmente, uma simulação para um sistema simplificado é apresentada (seção 5.3).

5.1 Formulação da função objetivo e das restrições para o DC-IHMPC com RTO

Graficamente, a arquitetura de controle para a integração RTO/DCMPC é análoga àquela para o caso centralizado e pode ser observada na Figura 9. Para M subsistemas, são gerados M valores de referência de entradas.

Como descrito na subseção 2.7.2 e posteriormente utilizado no DC-IHMPC apresentado na seção 4.1, a função objetivo deve ser uma combinação convexa entre funções quadráticas para cada subsistema. Isso é representado, para M subsistemas, pelas equações (82a) e (82b).

Figura 9: Estrutura de controle: Integração RTO/DCMPC

Fonte: elaboração própria (2020).

Quanto a cada função quadrática V_{ki} , estas são definidas de maneira análoga à equação (64), de modo que:

$$\begin{aligned}
 V_{ki} = & \sum_{l=1}^{\infty} \left\| \mathbf{y}_i(k+l|k) - \mathbf{y}_{ki}^{sp} - \boldsymbol{\delta}_{kii}^y - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{k iq}^y \right\|_{\mathbf{Q}_{yi}}^2 + \\
 & \sum_{l=0}^{\infty} \left\| \mathbf{u}_i(k+l|k) - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{ki}^u \right\|_{\mathbf{Q}_{ui}}^2 + \\
 & \sum_{l=0}^m \left\| \Delta \mathbf{u}_i(k+l|k) \right\|_{\mathbf{R}_i}^2 + \sum_{q=1}^M \left\| \boldsymbol{\delta}_{k iq}^y \right\|_{\mathbf{S}_{yi}}^2 + \left\| \boldsymbol{\delta}_{ki}^u \right\|_{\mathbf{S}_{ui}}^2
 \end{aligned} \tag{158}$$

Assim como para o caso centralizado, $\mathbf{y}_{ki}^{sp}$ não é mais uma constante previamente estipulada, mas sim uma variável a ser calculada. Esta deve estar entre um valor máximo ($\mathbf{y}^{max}$) e um valor mínimo ($\mathbf{y}^{min}$). O parâmetro que passa a ser o alvo, ou “target” é $\mathbf{u}_{ki}^{des}$. Utilizam-se também variáveis de folga tanto na soma para as entradas quanto para as saídas: $\boldsymbol{\delta}_{ki}^u$ e $\boldsymbol{\delta}_{ki}^y$.

Para a primeira soma infinita, a situação é a mesma do controlador definido na seção 4.1: divide-se a mesma em duas partes. Na primeira parte soma-se até m , que é o horizonte

de controle e na segunda, até o infinito. Assim como antes, o modelo de estado utilizado é o OPOM e o desenvolvimento é o mesmo das equações (98) - (103b). Já para a segunda soma infinita, define-se a seguinte restrição terminal para a função objetivo:

$$\mathbf{u}_i(k+l|k) - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{ki}^u = \mathbf{0} \quad l \geq m \quad (159)$$

Como $\Delta \mathbf{u}_i(k+l|k) = \mathbf{0} \quad l \geq m$ também é uma restrição, basta que $\mathbf{u}_i(k+l|k)$, $l = m$ obedeça à restrição (159). Esta por sua vez, pode ser escrita como:

$$\mathbf{u}_i(k-1) + \tilde{\mathbf{D}}_i^u \Delta \mathbf{u}_{ki} - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{ki}^u = \mathbf{0} \quad (160)$$

Em que¹:

$$\tilde{\mathbf{D}}_i^u = \underbrace{[\mathbf{I}_{nu_i} \cdots \mathbf{I}_{nu_i}]}_m \quad (161)$$

Assim, pode-se reescrever a segunda soma infinita da seguinte forma:

$$\sum_{l=0}^{m-1} \|\mathbf{u}_i(l+l|k) - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{ki}^u\|_{\mathbf{Q}_{ui}}^2 = \|\bar{\mathbf{I}}_{nu_i} (\mathbf{u}_i(k-1) - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{ki}^u) + \mathbf{D}_{mi}^u \Delta \mathbf{u}_{ki}\|_{\mathbf{Q}_{gi}^u}^2 \quad (162)$$

Em que:

$$\bar{\mathbf{I}}_{nu_i} = \begin{bmatrix} \mathbf{I}_{nu_i} \\ \mathbf{I}_{nu_i} \\ \vdots \\ \mathbf{I}_{nu_i} \end{bmatrix} \quad (163) \quad \mathbf{D}_{mi}^u = \begin{bmatrix} \mathbf{I}_{nu_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I}_{nu_i} & \mathbf{I}_{nu_i} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{nu_i} & \mathbf{I}_{nu_i} & \cdots & \mathbf{I}_{nu_i} \end{bmatrix} \quad (164)$$

$$\mathbf{Q}_{gi}^u = \text{diag} \left(\underbrace{\mathbf{Q}_{ui} \cdots \mathbf{Q}_{ui}}_m \right) \quad (165)$$

Dessa forma, V_k pode ser escrita como:

¹Lembre que nu_i refere-se ao número de entradas compreendidas dentro do subsistema i

$$\begin{aligned}
V_k = & \sum_{q=1}^M w_q \left\{ \left\| \bar{\mathbf{I}}_q \left[\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \sum_{j=1}^M \delta_{kqj}^y \right] + \Psi_{\mathbf{g}q} \mathbf{F}_{\mathbf{x}q} \mathbf{x}_q^d(k) + \right. \right. \\
& \sum_{j=1}^M \left[(\mathbf{D}_{\mathbf{m}qj}^0 + \Psi_{\mathbf{g}q} \mathbf{F}_{\mathbf{u}qj}) \Delta \mathbf{u}_{kj} \right] \left\|_{\mathbf{Q}_{\mathbf{g}q}^y}^2 + \left\| \mathbf{F}_{\mathbf{x}q} \mathbf{x}_q^d(k) + \sum_{j=1}^M \{ \mathbf{F}_{\mathbf{u}qj} \Delta \mathbf{u}_{kj} \} \right\|_{\bar{\mathbf{Q}}_{\mathbf{g}q}}^2 + \right. \\
& \left\| \bar{\mathbf{I}}_{nu_q} [\mathbf{u}_q(k-1) - \mathbf{u}_{kq}^{des} - \delta_{kq}^u] + \mathbf{D}_{\mathbf{m}q}^u \Delta \mathbf{u}_{kq} \right\|_{\mathbf{Q}_{\mathbf{g}q}^u}^2 + \\
& \left. \left\| \Delta \mathbf{u}_{kq} \right\|_{\mathbf{R}_{\mathbf{g}q}}^2 + \sum_{j=1}^M \left\| \delta_{kqj}^y \right\|_{\mathbf{S}_q^y}^2 + \left\| \delta_{kq}^u \right\|_{\mathbf{S}_q^u}^2 \right\} \quad (166)
\end{aligned}$$

Observe que há M^2 variáveis de folga para as saídas, mas somente M variáveis de folga para as entradas. Ou seja, não há variáveis de folga cruzadas entre uma saída e outra. Isso já era esperado, já que $\Delta \mathbf{u}_{kj}$, $j \neq i$, não influencia $\mathbf{u}_i$. Finalmente, define-se o problema de otimização para o subsistema i como:

$$\min_{\Delta \mathbf{u}_{ki}, \mathbf{y}_{ki}^{sp}, \bar{\delta}_{ki}, \delta_{ki}^u} V_k \quad (167)$$

$$- \Delta \mathbf{u}_i^{max} \leq \Delta \mathbf{u}_i(k+l) \leq \Delta \mathbf{u}_i^{max} \quad \forall l \quad (168)$$

$$\mathbf{y}_i^{min} \leq \mathbf{y}_{ki}^{sp} \leq \mathbf{y}_i^{max} \quad (169)$$

$$\Delta \mathbf{u}_i(k+l|k) = \mathbf{0}, \quad l \geq m \quad (170)$$

$$\mathbf{u}_i^{min} \leq \mathbf{u}_i(k-1) + \sum_{q=0}^l \Delta \mathbf{u}_i(k+q) \leq \mathbf{u}_i^{max} \quad l = 0 \dots m-1 \quad (171)$$

$$\mathbf{u}_i(k-1) + \tilde{\mathbf{D}}_i^u \Delta \mathbf{u}_{ki} - \mathbf{u}_{ki}^{des} - \delta_{ki}^u = \mathbf{0} \quad (172)$$

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \delta_{kqi}^y + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj} - \delta_{kqj}^y \right\} = \mathbf{0} \quad q = 1, \dots, M \quad (173)$$

Cabe lembrar: as variáveis de folga relacionadas às saídas foram agrupadas em um vetor $\bar{\delta}_{ki}$. Além disso, os controladores repetem o cálculo de (167) - (173) para todos os subsistemas i . Observe que, para cada subsistema i , (172) e (173) podem ser usadas para eliminar as variáveis de folga, de modo que $\Delta \mathbf{u}_{ki}$ e $\mathbf{y}_{ki}^{sp}$ são as verdadeiras variáveis de decisão do sistema.

Assim como para o controlador do Capítulo 4, pode-se escrever a função objetivo deste capítulo em uma forma matricial para programação em Matlab. Esta pode ser vista no Apêndice B.

5.1.1 Cálculo iterativo para obtenção de solução ótima

Assim como para o controlador apresentado no Capítulo 4, a função objetivo V_k é uma combinação convexa de funções quadráticas e apresenta, portanto, solução única. Entretanto, o controlador i não dispõe dos valores de $\Delta \mathbf{u}_{kj}$, $\mathbf{y}_{kj}^{sp}$, $\bar{\delta}_{kj}$ e δ_{kj}^u , $j \neq i$, já que todas as variáveis são calculadas por todos os controladores simultaneamente. Dessa forma, é necessário que o método seja feito de modo iterativo para que as respostas obtidas se aproximem da solução global.

O seguinte algoritmo, análogo ao Algoritmo 1 apresentado na subseção 4.1.1, deve ser utilizado para garantir que o controlador com RTO convirja para a solução ótima. Mais uma vez, se o número de iterações não for grande o bastante, é obtida uma solução chamada “sub ótima”, que também tem a capacidade de levar a planta para o estado desejado.

Quanto à notação, utilizou-se a mesma que na subseção 4.1.1, a saber: p é a iteração atual; $\Delta \mathbf{u}_{ki}^{(p*)}$, $\mathbf{y}_{ki}^{sp(p*)}$, $\bar{\delta}_{ki}^{(p*)}$ e $\delta_{ki}^{u(p*)}$ é a solução obtida pelo controlador do subsistema i ao resolver o problema (167) - (173); $\Delta \mathbf{u}_{ki}^{(p)}$, $\mathbf{y}_{ki}^{sp(p)}$, $\bar{\delta}_{ki}^{(p)}$ e $\delta_{ki}^{u(p)}$ é a solução final para a iteração p .

Algoritmo 2:

1. Defina critério de parada (número fixo de iterações ou norma do erro de aproximação)
2. Estabeleça $p = 1$
3. Defina $\Delta \mathbf{u}_{ki}^{(p-1)} = [\Delta \mathbf{u}_i^T(k|k-1) \cdots \Delta \mathbf{u}_i^T(k+m-2|k-1) \mathbf{0}^T]^T$, $\mathbf{y}_{ki}^{sp(p-1)} = \mathbf{y}_{k-1i}^{sp}$
 $\bar{\delta}_{ki} = \bar{\delta}_{k-1i}$, $\delta_{ki}^{u(p-1)} = \delta_{k-1i}^u$
4. Enquanto o critério de parada não for atingido, faça:
 - (a) Avalie: $\left(\Delta \mathbf{u}_{ki}^{(p*)}, \mathbf{y}_{ki}^{sp(p*)}, \bar{\delta}_{ki}^{(p*)}, \delta_{ki}^{u(p*)} \right) = \arg \min V_k \left(\mathbf{x}(k), \Delta \mathbf{u}_{kj}^{(p-1)}, \mathbf{y}_{kj}^{sp(p-1)}, \bar{\delta}_{kj}^{(p-1)}, \delta_{kj}^{u(p-1)}, j \neq i \right) \forall i$ sujeito a (168) - (173)
 - (b) Para $i = 1, \dots, M$, estabeleça:

$$\begin{bmatrix} \Delta \mathbf{u}_{ki}^{(p)} \\ \mathbf{y}_{ki}^{sp(p)} \\ \bar{\delta}_{ki}^{(p)} \\ \delta_{ki}^{u(p)} \end{bmatrix} \leftarrow w_i \begin{bmatrix} \Delta \mathbf{u}_{ki}^{(p*)} \\ \mathbf{y}_{ki}^{sp(p*)} \\ \bar{\delta}_{ki}^{(p*)} \\ \delta_{ki}^{u(p*)} \end{bmatrix} + (1 - w_i) \begin{bmatrix} \Delta \mathbf{u}_{ki}^{(p-1)} \\ \mathbf{y}_{ki}^{sp(p-1)} \\ \bar{\delta}_{ki}^{(p-1)} \\ \delta_{ki}^{u(p-1)} \end{bmatrix}$$

- (c) Estabeleça $p = p + 1$
- 5. Implemente $\Delta \mathbf{u}_i(k|k), \forall i$
- 6. Mova para o próximo instante e retorne para 2.

Mais uma vez, os valores de w_i são os mesmos utilizados na combinação convexa para a função objetivo. A cada iteração, a função objetivo V_k para o subsistema i recebe os novos valores das variáveis calculadas pelos outros subsistemas $j \neq i$ e os utiliza como parâmetros. Quanto ao critério de parada, o Algoritmo 2 pode ser repetido até um número predeterminado de iterações, ou então até que $\left\| \Delta \mathbf{u}_{ki}^{(p)} - \Delta \mathbf{u}_{ki}^{(p-1)} \right\|^2 \leq \epsilon$, no qual ϵ é um número pequeno.

Quanto à convergência dos valores da função objetivo e das variáveis para o ponto ótimo global, a prova é idêntica àquela apresentada na subseção 4.2.2. A inclusão das variáveis $\mathbf{y}_{ki}^{sp}$ e δ_{ki}^u não altera o cálculo realizado para os Teoremas 1 e 2. Assim, as provas não serão repetidas nesta seção.

5.2 Factibilidade recursiva e estabilidade do problema proposto com RTO

Como mencionado na seção anterior, a convergência do Algoritmo 2 não foi discutida, pois o cálculo é análogo ao realizado para o Algoritmo 1. Ou seja, dada a existência de uma solução ótima global representada por $\Delta \mathbf{u}_{ki}^*, \mathbf{y}_{ki}^{sp*}, \bar{\delta}_{ki}^*, \delta_{ki}^{u*}$ e um valor mínimo para V_k dado por $V_k^* = V_k(\Delta \mathbf{u}_{ki}^*, \mathbf{y}_{ki}^{sp*}, \bar{\delta}_{ki}^*, \delta_{ki}^{u*})$ (o estado inicial foi omitido), então para p grande o suficiente, o Algoritmo 2 converge para tais valores.

A questão da convexidade também não precisa ser discutida novamente. Isso é porque o problema definido em (167) - (173) também é um problema quadrático apresentando ponto de mínimo global. Quanto ao conjunto de todas as possíveis soluções, denominado na subseção 4.2.1 genericamente por $\mathcal{U}^{nu}$, este também se mantém convexo. Apesar da adição de uma nova restrição de igualdade (a saber, (172)), há também a adição de uma nova variável de folga, sendo esta irrestrita. O conjunto dos possíveis valores de $\mathbf{y}_{ki}^{sp}$ também é convexo, pois os possíveis valores são números reais com limites máximo e mínimo. Entretanto, cabe falar sobre a factibilidade recursiva e a estabilidade do problema, bem como cabe falar sobre as restrições e sua relação com o modelo iterativo.

5.2.1 A questão das restrições e o modelo iterativo com RTO

Assim como na subseção 4.2.4, cabe mostrar que as restrições são obedecidas na iteração p , desde que em $p - 1$ elas tenham sido obedecidas. Para as restrições (168), (169) e (171), o cálculo é o mesmo daquele apresentado na subseção 4.2.4 e não será repetido aqui. A restrição (168) também é inerente à programação do controlador e não requer prova matemática.

Quanto às restrições (172) e (173), essas são, como já mencionado, decorrentes de aprimoramentos em relação a controladores já pertencentes à literatura. Cabe agora falar sobre estas restrições com relação ao método iterativo.

Para (172), seja a solução obtida pelo controlador i no instante p ao resolver o Algoritmo 2 tal que:

$$\mathbf{u}_i(k-1) + \tilde{\mathbf{D}}_i^u \Delta \mathbf{u}_{ki}^{(p*)} - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{ki}^{u(p*)} = \mathbf{0} \quad (174)$$

Suponha que tal restrição foi obedecida na iteração anterior, então:

$$\mathbf{u}_i(k-1) + \tilde{\mathbf{D}}_i^u \Delta \mathbf{u}_{ki}^{(p-1)} - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{ki}^{u(p-1)} = \mathbf{0} \quad (175)$$

Multiplicando (174) por w_i e (175) por $(1 - w_i)$ e somando as duas chega-se à expressão válida em p :

$$\begin{aligned} \mathbf{u}_i(k-1) + \tilde{\mathbf{D}}_i^u \left[w_i \Delta \mathbf{u}_{ki}^{(p*)} + (1 - w_i) \Delta \mathbf{u}_{ki}^{(p-1)} \right] - \mathbf{u}_{ki}^{des} - w_i \boldsymbol{\delta}_{ki}^{u(p*)} + \\ (1 - w_i) \boldsymbol{\delta}_{ki}^{u(p-1)} = \mathbf{0} \Leftrightarrow \mathbf{u}_i(k-1) + \tilde{\mathbf{D}}_i^u \Delta \mathbf{u}_{ki}^{(p)} - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{ki}^{u(p)} = \mathbf{0} \end{aligned} \quad (176)$$

Quanto a (173), o raciocínio é o mesmo daquele apresentado na subseção 4.2.4, mas foi repetido aqui para o caso com RTO para mostrar como a variável $\mathbf{y}_{ki}^{sp}$ participa do cálculo. A cada iteração, o controlador do subsistema i resolve o problema dado por (167) - (173). Assuma que (173) é válida em $p - 1$:

$$\begin{aligned} \mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp(p-1)} - \boldsymbol{\delta}_{kqi}^{(p-1)} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki}^{(p-1)} + \\ \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kqj}^{(p-1)} \right\} = \mathbf{0} \end{aligned} \quad q = 1, \dots, M \quad (177)$$

Seja agora a solução calculada na iteração p para o subsistema i (Algoritmo 1 passo 4.(a)) tal que:

$$\mathbf{x}_i^s(k) - \mathbf{y}_{ki}^{sp(p^*)} - \boldsymbol{\delta}_{kii}^{(p^*)} + \tilde{\mathbf{D}}_{ii}^0 \Delta \mathbf{u}_{ki}^{(p^*)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{ij}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kij}^{(p-1)} \right\} = \mathbf{0} \quad (178a)$$

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp(p-1)} - \boldsymbol{\delta}_{kqi}^{(p^*)} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki}^{(p^*)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kqj}^{(p-1)} \right\} = \mathbf{0} \quad q \neq i \quad (178b)$$

Então, multiplique (177) por $(1 - w_q)$, (178a) por w_i e (178b) por w_q e some cada equação em (177) com as equações (178a) e (178b) obedecendo aos índices corretamente. Dessa forma, obtêm-se:

$$\mathbf{x}_i^s(k) - \mathbf{y}_{ki}^{sp(p)} - \boldsymbol{\delta}_{kii}^{(p)} + \tilde{\mathbf{D}}_{ii}^0 \Delta \mathbf{u}_{ki}^{(p)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{ij}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kij}^{(p-1)} \right\} = \mathbf{0} \quad (179a)$$

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp(p-1)} - \boldsymbol{\delta}_{kqi}^{(p)} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki}^{(p)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj}^{(p-1)} - \boldsymbol{\delta}_{kqj}^{(p-1)} \right\} = \mathbf{0} \quad q \neq i \quad (179b)$$

Note que (179a) e (179b) são as M equações para o subsistema i , mas o mesmo é válido para todos os outros subsistemas. Assim, repetindo esta mesma conta (177) - (179b) para os outros subsistemas e depois somando cada equação (179a) e (179b) de cada subsistema com a equação de mesmo índice para os outros subsistemas, chega-se a:

$$\begin{aligned}
& M\mathbf{x}_q^s(k) - \left[(M-1)\mathbf{y}_{kq}^{sp(p-1)} + \mathbf{y}_{kq}^{sp(p)} \right] - \left[(M-1)\boldsymbol{\delta}_{kqi}^{(p-1)} + \boldsymbol{\delta}_{kqi}^{(p)} \right] + \\
& \tilde{\mathbf{D}}_{qi}^0 \left[(M-1)\Delta\mathbf{u}_{ki}^{(p-1)} + \Delta\mathbf{u}_{ki}^{(p)} \right] + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \left[(M-1)\Delta\mathbf{u}_{kj}^{(p-1)} + \Delta\mathbf{u}_{kj}^{(p)} \right] - \right. \\
& \left. \left[(M-1)\boldsymbol{\delta}_{kqj}^{(p-1)} + \boldsymbol{\delta}_{kqj}^{(p)} \right] \right\} = \mathbf{0} \quad q = 1, \dots, M
\end{aligned} \tag{180}$$

Finalmente, subtraindo (177) de (180) $(M-1)$ vezes, obtêm-se:

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp(p)} - \boldsymbol{\delta}_{kqi}^{(p)} + \tilde{\mathbf{D}}_{qi}^0 \Delta\mathbf{u}_{ki}^{(p)} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta\mathbf{u}_{kj}^{(p)} - \boldsymbol{\delta}_{kqj}^{(p)} \right\} = \mathbf{0} \quad q = 1, \dots, M \tag{181}$$

que é (177) em p .

5.2.2 Factibilidade recursiva do controlador proposto com RTO

A factibilidade recursiva, assim como na subsecção 4.2.5 foi discutida na forma de um Teorema. Tal assunto foi discutido tendo-se em mente que, para um número suficientemente grande de iterações, o Algoritmo 2 converge para a solução ótima. A prova entretanto, é válida mesmo para o caso em que uma solução sub ótima seja obtida. como pelo menos uma das duas é sempre o caso, a factibilidade recursiva é garantida.

Para o Teorema e as provas a seguir, tenha em mente que o problema de minimização da função V_k é convexo sobre um conjunto convexo e tem solução única.

Teorema 6 - Para um sistema estável, modelo nominal, não perturbado, se o controlador definido através das equações (167) - (173) é factível no instante k , ele permanece factível nos instantes subsequentes.

Prova: utilizando-se da mesma notação das seções anteriores, suponha então que $\Delta\mathbf{u}_{ki}^*$, y_{ki}^{sp*} , $\bar{\boldsymbol{\delta}}_{ki}^*$, $\boldsymbol{\delta}_{ki}^{u*}$ seja a solução ótima (e portanto global) para o problema de o controlador i minimizar V_k no instante de tempo k . Assim, o movimento $\Delta\mathbf{u}_i^*(k|k)$ é injetado no

sistema no instante k para todos os subsistemas $i = 1, \dots, M$.

Considere a seguir a sequência: $\Delta \tilde{\mathbf{u}}_{k+1i} = [\Delta \mathbf{u}_i^{*T}(k+1|k) \Delta \mathbf{u}_i^{*T}(k+2|k) \dots \Delta \mathbf{u}_i^{*T}(k+m-1|k) \mathbf{0}^T]^T$, $\tilde{\mathbf{y}}_{ki}^{sp} = \mathbf{y}_{ki}^{sp*}$, $\tilde{\boldsymbol{\delta}}_{k+1i} = \tilde{\boldsymbol{\delta}}_{ki}^*$, $\tilde{\boldsymbol{\delta}}_{k+1i}^u = \boldsymbol{\delta}_{ki}^{u*}$, $i = 1, \dots, M$. É trivial mostrar que tal sequência obedece às restrições (168) - (171), já que $\mathbf{0} \in \mathcal{U}^{nu}$.

Com relação às equações (172) e (173), considere as seguintes expressões para qualquer subsistema i e a sequência acima. Então:

$$\mathbf{u}_i(k) + \tilde{\mathbf{D}}_i^u \Delta \tilde{\mathbf{u}}_{k+1i} - \mathbf{u}_{ki}^{des} - \tilde{\boldsymbol{\delta}}_{k+1i}^u \quad (182a)$$

$$\mathbf{x}_q^s(k+1) - \tilde{\mathbf{y}}_{k+1q}^{sp} - \tilde{\boldsymbol{\delta}}_{k+1qi}^y + \tilde{\mathbf{D}}_{qi}^0 \Delta \tilde{\mathbf{u}}_{k+1i} + \sum_{\substack{j=1 \\ j \neq i}}^M \tilde{\mathbf{D}}_{qj}^0 \Delta \tilde{\mathbf{u}}_{k+1j} - \tilde{\boldsymbol{\delta}}_{k+1qj}^y \quad q = 1, \dots, M \quad (182b)$$

Entretanto, $\mathbf{u}_i(k) = \mathbf{u}_i(k-1) + \Delta \mathbf{u}_i^*(k|k)$ e também $\mathbf{x}_i^s(k+1) = \mathbf{x}_i^s(k) + \sum_{j=1}^M \mathbf{D}_{ij}^0 \Delta \mathbf{u}_j^*(k|k)$, $i = 1, \dots, M$. Substituindo-se estes resultados nas expressões acima, obtêm-se os lados esquerdos de (172) e (173), que são nulos. Ou seja, $\Delta \tilde{\mathbf{u}}_{k+1i}$, $\tilde{\mathbf{y}}_{ki}^{sp}$, $\tilde{\boldsymbol{\delta}}_{k+1i}^*$, $\tilde{\boldsymbol{\delta}}_{k+1i}^u$ é uma solução factível. Havendo pelo menos uma solução factível para $k+1$, pode-se repetir o mesmo raciocínio recursivamente para $k+2$ e assim por diante, de modo que a factibilidade recursiva está provada. $\square$

5.2.3 Estabilidade do controlador proposto com RTO

Assim como na subseção 4.2.6, cabe mostrar que a função objetivo diminui de cada instante k para $k+1$. Com isso, prova-se que o controlador é estável e que é capaz de levar uma determinada planta a qualquer set-point desejado, desde que o mesmo seja factível. Mais uma vez, isso vai ser feito na forma de Teorema.

Teorema 7 - Para um sistema estável, modelo nominal, não perturbado, se o estado estacionário desejado é atingível, o controlador produzido pela solução do problema definido pelas equações (167) - (173) (com matrizes de variáveis de folga $\mathbf{S}_y$ e $\mathbf{S}_u$ positivas e definidas) é assintoticamente estável. As entradas do sistema são levadas para os set-points desejados e as saídas do sistema se mantêm dentro de faixas pré-estabelecidas. Se o set-point desejado não for atingível, o controlador deve estabilizar o sistema em um

estado estacionário que minimize a distância do mesmo para o estado desejado.

Prova: dado $V_k^* = V_k(\Delta \mathbf{u}_{ki}^*, \mathbf{y}_{ki}^{sp*}, \bar{\boldsymbol{\delta}}_{ki}^*, \boldsymbol{\delta}_{ki}^u)$ o valor da função objetivo obtido pelo controlador i em k após um número suficientemente grande de iterações p (Algoritmo 2) considerando (167) - (173). Seja $\tilde{V}_{k+1}$ a solução obtida pela sequência $\Delta \tilde{\mathbf{u}}_{k+1i}, \tilde{\mathbf{y}}_{ki}^{sp}, \tilde{\boldsymbol{\delta}}_{k+1i}, \tilde{\boldsymbol{\delta}}_{k+1i}^u$ definida na subseção 5.2.2, para este mesmo controlador. É fácil observar que:

$$\begin{aligned} \tilde{V}_{k+1} - V_k^* = \sum_{q=1}^M w_q \left\{ - \left\| \mathbf{y}_q(k|k) - \mathbf{y}_{kq}^{sp*} - \sum_{j=1}^M \boldsymbol{\delta}_{kqj}^* \right\|_{\mathbf{Q}_q}^2 \right\} - \\ \left\| \mathbf{u}_i^*(k|k) - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{kq}^{u*} \right\|_{\mathbf{Q}_{uq}}^2 - \left\| \Delta \mathbf{u}_i^*(k|k) \right\|_{\mathbf{R}_i}^2 \end{aligned} \quad (183)$$

Como as matrizes $\mathbf{Q}_q, \mathbf{R}_q$ são positivas, enquanto $\mathbf{y}_q(k|k) - \mathbf{y}_{kq}^{sp*} - \sum_{j=1}^M \boldsymbol{\delta}_{kqj}^*, \mathbf{u}_i^*(k|k) - \mathbf{u}_{ki}^{des} - \boldsymbol{\delta}_{kq}^{u*}$ ou $\Delta \mathbf{u}_i^*(k|k)$ não forem nulos:

$$\tilde{V}_{k+1} - V_k^* \leq 0 \Leftrightarrow \tilde{V}_{k+1} \leq V_k^* \quad (184)$$

Consequentemente:

$$V_{k+1}^* \leq V_k^* \quad (185)$$

Tal situação se repete para $k+1, k+2$ e assim sucessivamente. Desta forma, fica mostrado que a função objetivo é monotonicamente decrescente. $\square$

5.3 Resultados de simulação para sistemas simplificados com RTO

O seguinte sistema 3×2 , proposto por González e Odloak [26], foi utilizado para o estudo preliminar antes de se simular o controlador distribuído para um caso industrial. Assim como no capítulo anterior, considerou-se modelo nominal e utilizou-se Filtro de Kalman conforme descrito na seção 3.3. Com relação ao método iterativo, o Algoritmo 2 descrito na subseção 5.1.1 foi utilizado.

$$\mathbf{G}(s) = \begin{bmatrix} \frac{0.25}{3.5s + 1} & \frac{.14}{2.77s + 1} \\ \frac{0.9}{25s + 1} & \frac{0.19s - 2.9}{19.73s^2 + 10.83s + 1} \\ \frac{1.25}{11.11s + 1} & \frac{-5}{12.91s + 1} \end{bmatrix} \quad (186)$$

Os parâmetros de sintonia foram:

$$\begin{aligned} \mathbf{Q}_y &= \text{diag}(1 \quad 1 \quad 1) & \Delta T &= 1 \text{ min} \\ \mathbf{Q}_u &= 0.5 \text{diag}(1 \quad 1) & p &= 2 \text{ e } 4 \\ \mathbf{R} &= 0.05 \text{diag}(1 \quad 1) & m &= 3 \\ \mathbf{S}^y &= 10^4 \text{diag}(1 \quad 1 \quad 1) & \mathbf{w} &= [0.5 \ 0.5]^T \\ \mathbf{S}^u &= 10^2 \text{diag}(1 \quad 1) \end{aligned}$$

Já os valores máximos e mínimos para as faixas nas saídas, bem como os valores iniciais encontram-se na Tabela 1:

Tabela 1: Valores mínimos, máximos e iniciais para as saídas - Sistema 3×2

Variável	y^{min}	y^{max}	y_0
y_1	530	560	549.5
y_2	580	750	704.27
y_3	650	900	690.62

Fonte: González e Odloak (2011) [26].

Finalmente, para as restrições e valores iniciais nas entradas vide Tabela 2.

Tabela 2: Valores mínimos, máximos e iniciais para as entradas - Sistema 3×2

Variável	Δu^{max}	u^{min}	u^{max}	u_0
u_1	25	75	250	230.6
u_2	25	25	101	60.26

Fonte: González e Odloak (2011) [26].

No início da simulação, foram feitas então as seguintes mudanças de alvo:

$$\mathbf{u}^{des} = [210 \quad 30]^T$$

A seguir, após 40 minutos (40 intervalos de simulação), foram alteradas as faixas nas saídas (vide Tabela 3), levando a uma situação na qual $\mathbf{u}^{des}$ passou a ser inatingível.

Tabela 3: Tabela com novos valores mínimos e máximos para as saídas - Sistema 3×2

Variável	y^{min}	y^{max}
y_1	520	550
y_2	500	620
y_3	500	550

Fonte: elaboração própria (2020).

Para os parâmetros acima, pareando-se (y_1, y_2, u_1) e (y_3, u_2) , os resultados foram bastante semelhantes a um sistema centralizado, e o controlador mostrou boa performance. O caso aqui apresentado é bem menos sensível à mudança do número de iterações do que o sistema 2×2 do capítulo anterior. Os resultados da simulação podem ser observados nas Figuras 10 a 12.

Figura 10: Simulação: saídas para um sistema 3×2 com RTO

Fonte: elaboração própria (2020).

Figura 11: Simulação: entradas para um sistema 3×2 com RTO

Fonte: elaboração própria (2020).

Figura 12: Função objetivo para um sistema 3×2 com RTO

Fonte: elaboração própria (2020).

Como esperado, enquanto o sistema era atingível, todos os controladores conseguiram levar os valores das entradas para seus alvos enquanto mantinham os valores das saídas dentro de suas faixas desejadas. O valor da função objetivo foi levado a zero em torno de 20 minutos para todos os controladores. A partir do momento em que o valor dos alvos

não era mais factível por conta de redução drástica nos valores das faixas para as saídas, os controladores mantiveram as saídas dentro de suas faixas, sendo que y_2 e y_3 foram mantidas ambas em seus limites superiores. Já os alvos nas entradas não foram atingidos, como era esperado e a função objetivo não vai a zero.

O controlador com $p = 4$ teve um desempenho mais parecido com o controlador centralizado do que o controlador com $p = 2$, mas o desempenho de ambos é muito bom. Quanto aos tempos de computação, o controlador centralizado levou 16.11ms em média a cada instante para computar a solução, já o controlador distribuído levou 10.64ms em média a cada instante para $p = 2$ (já considerando um controlador por servidor) e 44.42ms em média a cada instante para $p = 4$. Ou seja, para $p = 2$, o tempo de computação é menor do que no caso do controlador centralizado e com um desempenho muito similar.

6 Proposta para o controlador DC-IHMPC com tempo morto

Esquemas de controle preditivo baseado em modelo que atuem junto a outras camadas de otimização, como o RTO e que possam operar em sistemas não quadrados (menos entradas que saídas) por conta, por exemplo, de operação com faixas, além de estarem bem definidos para o caso centralizado (e agora também para o caso distribuído), são muito próximos aos casos industriais. Em alguns casos entretanto, os sistemas de interesse têm tempo morto, o que ainda não foi abordado para o caso distribuído. Nesta seção, foi discutido o DC-IHMPC com tempo morto. Tal formulação é uma extensão do caso descrito no Capítulo 5, pois atua junto a RTO e possui faixas de operação nas saídas e alvos nas entradas. Com isso, os controladores distribuídos aqui definidos abrangem uma vasta gama de situações encontradas em processos industriais.

6.1 Formulação da função objetivo e das restrições para o DC-IHMPC com tempo morto

Considerando que, assim como no caso centralizado, a função objetivo para o caso com tempo morto é a mesma já utilizada para controladores com RTO sem tempo morto (seção 5.1 - equação (166)), então basta mostrar aqui como manipular tal função e que diferenças utilizar nas restrições para um controlador efetivo.

Começando pelas restrições, (168) - (172) se mantêm inalteradas. Já a equação (173) (restrição de igualdade que mantém restritas as somas infinitas nas saídas de V_k para todos os subsistemas), a diferença é que, para qualquer relação entrada/saída que possua como tempo morto $\theta_{\max}^i$, $\forall i$, uma ação no instante k tem efeito somente em $k + \theta_{\max}^i$.² Logo, a restrição é escrita como:

²Não confundir $\theta_{\max}$ com $\theta_{\max}^i$. O primeiro deles é utilizado na definição do estado e refere-se ao maior tempo morto do sistema todo, o segundo é utilizado na formulação do controlador e refere-se ao maior tempo morto de um determinado subsistema i .

$$\mathbf{x}_q^s(k + m + \theta_{\max}^q | k) - \mathbf{y}_{kq}^{sp} - \boldsymbol{\delta}_{kqi}^y - \sum_{\substack{j=1 \\ j \neq q}}^M \boldsymbol{\delta}_{kqj}^y = \mathbf{0} \quad q = 1, \dots, M \quad (187)$$

Para manipular $\mathbf{x}_q^s(k + m + \theta_{\max}^q | k)$, utilizam-se as equações de estado descritas na seção 3.2. Para manter a notação tal qual Martins e Odloak [23], pode-se reescrever a função objetivo V_k da seguinte forma:

$$\begin{aligned} V_k = \sum_{q=1}^M w_q \left\{ \left\| \bar{\mathbf{A}}_q \mathbf{x}_q(k) + \bar{\mathbf{B}}_{qi} \Delta \mathbf{u}_{ki} + \sum_{\substack{j=1 \\ j \neq i}}^M \bar{\mathbf{B}}_{qj} \Delta \mathbf{u}_{kj} - \bar{\mathbf{I}}_q \left[\mathbf{y}_{kq}^{sp} + \boldsymbol{\delta}_{kqi}^y + \sum_{\substack{j=1 \\ j \neq i}}^M \boldsymbol{\delta}_{kqj}^y \right] \right\|_{\mathbf{Q}_{gq}}^2 + \right. \\ \left\| \mathbf{N}_{dq} \mathbf{A}_q^{\theta_{\max}^q} \left[\mathbf{A}_q^m \mathbf{x}_q(k) + \mathbf{C}_{oqi} \Delta \mathbf{u}_{ki} + \sum_{\substack{j=1 \\ j \neq i}}^M \mathbf{C}_{oqj} \Delta \mathbf{u}_{kj} \right] \right\|_{\bar{\mathbf{Q}}_{gq}}^2 + \\ \left\| \bar{\mathbf{I}}_{nu_q} [\mathbf{u}_q(k-1) - \mathbf{u}_{kq}^{des} - \boldsymbol{\delta}_{kq}^u] + \mathbf{D}_{mq}^u \Delta \mathbf{u}_{kq} \right\|_{\mathbf{Q}_{gq}^u}^2 + \\ \left. \left\| \Delta \mathbf{u}_{kq} \right\|_{\mathbf{R}_{gq}}^2 + \sum_{j=1}^M \left\| \boldsymbol{\delta}_{kqj}^y \right\|_{\mathbf{S}_q^y}^2 + \left\| \boldsymbol{\delta}_{kq}^u \right\|_{\mathbf{S}_q^u}^2 \right\} \quad (188) \end{aligned}$$

Em que:

$$\bar{\mathbf{A}}_i = \begin{bmatrix} \mathbf{C}_i \\ \mathbf{C}_i \mathbf{A}_i \\ \vdots \\ \mathbf{C}_i \mathbf{A}_i^m \\ \vdots \\ \mathbf{C}_i \mathbf{A}_i^{m+\theta_{\max}^i} \end{bmatrix} \quad (189)$$

$$\bar{\mathbf{B}}_{ij} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{C}_i \mathbf{B}_{ij} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_i \mathbf{A}_i^{m-1} \mathbf{B}_{ij} & \mathbf{C}_i \mathbf{A}_i^{m-2} \mathbf{B}_{ij} & \dots & \mathbf{C}_i \mathbf{B}_{ij} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_i \mathbf{A}_i^{m-1+\theta_{\max}^i} \mathbf{B}_{ij} & \mathbf{C}_i \mathbf{A}_i^{m-2+\theta_{\max}^i} \mathbf{B}_{ij} & \dots & \mathbf{C}_i \mathbf{A}_i^{\theta_{\max}^i} \mathbf{B}_{ij} \end{bmatrix} \quad (190)$$

$$\mathbf{C}_{oij} = \begin{bmatrix} \mathbf{A}_i^{m-1} \mathbf{B}_{ij} & \mathbf{A}_i^{m-2} \mathbf{B}_{ij} & \cdots & \mathbf{A}_i \mathbf{B}_{ij} & \mathbf{B}_{ij} \end{bmatrix} \quad (191)$$

$$\mathbf{N}_{dq} = \begin{bmatrix} \mathbf{0}_{nu \cdot ny_i \cdot na_i \times ny_i} & \mathbf{I}_{nu \cdot ny_i \cdot na_i} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \quad (192)$$

Observe que nu não é função de cada subsistema, mas sim do sistema total (isso tem a ver como o estado foi definido na seção 3.2). Assim, cada controlador ($i = 1, \dots, M$) resolve o problema descrito pelas equações (167) - (172) e (187). Para garantir a convergência, utiliza-se também o modelo iterativo descrito pelo Algoritmo 2 (subseção 5.1.1). Em termos de notação, V_k pode ser reescrita como (188), embora isso não seja obrigatório. Finalmente, as questões de convexidade, factibilidade recursiva e estabilidade também são as mesmas já discutidas nas seções 4.2 e 5.2 e não foram rediscutidas aqui.

6.2 Resultados de simulação para sistemas simplificados com tempo morto

Para testar a adição de tempo morto ao sistema, utilizou-se o segundo sistema listado em González e Odloak [26]. Observe que este é o mesmo sistema utilizado na seção 5.3, porém com valores de tempo morto associados a todas as combinações de variáveis. Assim como no capítulo anterior, considerou-se modelo nominal e utilizou-se Filtro de Kalman conforme descrito na seção 3.3. Com relação ao método iterativo, o Algoritmo 2 descrito na subseção 5.1.1 foi utilizado.

$$\mathbf{G}(s) = \begin{bmatrix} \frac{0.25e^{-2s}}{3.5s + 1} & \frac{0.14e^{-5s}}{2.77s + 1} \\ \frac{0.9e^{-3s}}{25s + 1} & \frac{(0.19s - 2.9)e^{-4s}}{19.73s^2 + 10.83s + 1} \\ \frac{1.25e^{-5s}}{11.11s + 1} & \frac{-5e^{-6s}}{12.91s + 1} \end{bmatrix} \quad (193)$$

Desta vez, associou-se y_2 a u_2 , de modo que a divisão adotada foi de dois subsistemas com as respectivas variáveis: (y_1, u_1) e (y_2, y_3, u_2) . As condições iniciais são as mesmas

da seção 5.3, inclusive com relação às faixas de operação. A única diferença é que a alteração de faixas de operação dos valores da Tabela 1 para os da Tabela 3 é feita após 100 minutos (100 períodos de simulação). Isso é feito por conta do sistema com tempo morto naturalmente demorar mais para responder a mudanças.

Com relação aos parâmetros de sintonia, utilizaram-se os mesmos daqueles já utilizados na seção 5.3. Os resultados podem ser vistos nas Figuras 13 a 15. A situação é bem parecida com o caso anterior, sem tempo morto. Mais uma vez, o desempenho de ambos os controladores distribuídos com relação às entradas e saídas foi similar ao desempenho do controlador centralizado, embora o caso $p = 4$ se comporte de maneira ainda mais próxima ao controlador centralizado do que o caso $p = 2$, o que pode ser observado pela proximidade das curvas na Figura 15.

Assim como para o caso do Capítulo 5, as saídas ficam dentro de suas zonas enquanto que as entradas são levadas aos valores desejados. Quando há saturação das saídas y_2 e y_3 , o controlador já não consegue mais manter as entradas em seus valores desejados (vide Figuras 13 e 14). A Figura 15, por sua vez, mostra que o custo vai para zero enquanto é possível levar as entradas para os valores desejados mantendo as saídas dentro de suas faixas e depois se estabiliza em um valor não nulo quando isso não é mais possível.

Figura 13: Simulação: saídas para um sistema 3×2 com tempo morto

Fonte: elaboração própria (2020).

Figura 14: Simulação: entradas para um sistema 3×2 com tempo morto

Fonte: elaboração própria (2020).

Figura 15: Função objetivo para um sistema 3×2 com tempo morto

Fonte: elaboração própria (2020).

Quanto aos tempos de computação, a mesma situação à anterior é observada: o controlador centralizado levou 17.75ms em média a cada instante para computar a solução, já o controlador distribuído levou 10.73ms em média a cada instante para $p = 2$ (já considerando o tempo dividido como se fosse um controlador por servidor) e 49.26ms em média

a cada instante para $p = 4$. Ou seja, para $p = 2$, o tempo de computação é menor do que no caso do controlador centralizado e com um desempenho muito similar.

7 Proposta para o controlador DC-IHMPC com diferentes escalas de tempo

Até aqui, foram feitas adaptações dos trabalhos de IHMPC com o modelo OPOM (incluindo integração com RTO, possibilidade de lidar com tempo morto, etc.) para funcionar de maneira distribuída e cooperativa. Como já comentado anteriormente, este sistema pode operar de maneira sub ótima e a descentralização possibilita implementar outros pacotes, como inteligência artificial, controle de falhas etc. A descentralização, pela sua flexibilidade intrínseca, possibilita também outros desenvolvimentos. Uma possibilidade de novo desenvolvimento, que foi desenvolvida integralmente neste trabalho, é a de operar em sistemas com múltiplas escalas de tempo.

Comumente, nas arquiteturas de MPCs, as variáveis de interesse são medidas em um determinado instante, chamado genericamente de k e passadas para o controlador. Este resolve o problema de otimização sujeito às restrições do sistema. O resultado é uma sequência de movimentos de controle dos quais o primeiro é aplicado ao sistema. A ação do controlador termina neste momento até que o tempo passe e chegue o próximo instante $k + 1$, assim recomeça-se todo o processo. O tempo decorrido entre k e $k + 1$ é o intervalo ou período de amostragem. Na maior parte dos trabalhos desenvolvidos na área de controladores preditivos baseados em modelo, utiliza-se um intervalo de amostragem fixo e pré-determinado.

Há, entretanto, sistemas em que algumas variáveis têm dinâmicas muito mais rápidas que outras, ficando difícil definir o intervalo de tempo entre k e $k + 1$. Estes são denominados sistemas com múltiplas escalas de tempo. A abordagem mais comum para lidar com tais sistemas é separar as diferentes variáveis em mais de uma arquitetura de controle independentes [61, 62], ou seja, é utilizada uma abordagem descentralizada. O controlador distribuído pode ser utilizado neste caso para lidar com tais sistemas por permitir uma arquitetura com descentralização sem ter as desvantagens de um sistema descentralizado. Até o presente momento, tal arranjo foi pouco abordado na literatura.

O que foi apresentado neste capítulo é a arquitetura de controle em que sub-sistemas distintos têm períodos de amostragem distintos (por exemplo, 1 minuto para o sistema 1 e 5 minutos para o sistema 2), de modo que as variáveis escolhidas para ficar dentro de um sub-sistema sejam controladas com período de amostragem diferente daquelas escolhidas para ficar em outro sub-sistema.

Como motivação para a existência deste tipo de controlador, cabe citar que é observada na indústria química uma complexidade muito grande de correntes e variáveis medidas. Tome, por exemplo, uma pequena parte de uma refinaria de petróleo: a destilação do óleo cru. Tal processo, constituído de poucos equipamentos, apresenta pelo menos oito variáveis manipuladas e dez variáveis controladas [19]. Em tal situação, diversos instrumentos de medição são utilizados para inferir dados sobre as inúmeras variáveis e permitir a realização do controle. Sendo assim, pode ser que pelo menos um destes instrumentos utilizados demore mais que os demais para realizar suas medições. Ou então, pode ser que uma das variáveis controladas tenha dinâmica muito mais lenta que as demais, sendo desnecessária, pouco eficiente ou mesmo impraticável a implementação de controle em intervalos de amostragem curtos.

Nestas situações, em que não se pode, ou não se tem interesse em medir todas as variáveis em um mesmo intervalo de tempo, não faz sentido que o controlador calcule trajetórias ótimas e que movimentos nas variáveis manipuladas sejam empregados sem as devidas medidas de seus efeitos. Com isso, fica clara a necessidade de um controlador capaz de realizar sua tarefa levando em conta intervalos de amostragem distintos para diferentes variáveis. O DC-IHMPC que foi apresentado até aqui neste trabalho se mostrou ideal para isto, pois a distribuição no tratar das variáveis facilita a tarefa de operar em intervalos de amostragem distintos.

7.1 As diferentes escalas de tempo

Considere um sistema no qual o período de amostragem para as variáveis não seja o mesmo para todas. Assuma também que tal sistema opere em um intervalo discreto de amostragem e que pode ser dividido em M sub-sistemas. Sendo assim, pode-se definir T_i o período de amostragem para o sub-sistema i . Assim, $T_f = 1$ é o período para o sistema mais rápido ou com maior amostragem em suas variáveis. Assume-se que os outros sub-sistemas terão períodos T_i tal que:

$$T_i = bT_f, \quad i = 1, \dots, M, \quad b \in \mathbb{N}^* \quad (194)$$

Finalmente, o horizonte de controle pode ser definido de maneira independente para cada sub-sistema e denominado m_i , $i = 1, \dots, M$.

7.2 Formulação da função objetivo e das restrições para o DC-IHMPC com múltiplas escalas de tempo

Para trabalhar com múltiplas escalas de tempo, define-se uma nova variável α_{kj} , denominada “filtro de instantes de amostragem”.

$$\alpha_{kj} = \begin{cases} 1 & k \bmod T_j = 0 \\ 0 & k \bmod T_j \neq 0 \end{cases} \quad j = 1, \dots, M \quad (195)$$

em que $k \bmod T_j$ representa o resto da divisão de k por T_j . Note que α_{kj} é similar a um Delta de Kronecker. Se em um determinado instante k o subsistema j está em seu período de amostragem, $\alpha_{kj} = 1$, se não $\alpha_{kj} = 0$. Claramente, para o subsistema mais rápido $\alpha_{kf} = 1, \forall k$.

Com relação às ações de controle, quando um controlador não está em seu período de amostragem, ou seja, $\alpha_{kj} = 0$ é desejável que $\Delta \mathbf{u}_j(k|k) = \mathbf{0}$, já que as saídas não reagem rapidamente a mudanças nas entradas ou então não se tem medidas acerca da variável fora de seu período de amostragem.

A função objetivo para um controlador distribuído e cooperativo que opera com set-points nas saídas e tem o mesmo número de variáveis controladas e manipuladas é a mesma utilizada no Capítulo 4 - equação (119). Esta deve ser resolvida mesmo quando um subsistema não está em seu período de amostragem para garantir que as variáveis de folga sejam atualizadas.

Dessa forma, pode definir-se o problema de otimização para o subsistema i como:

$$\min_{\Delta \mathbf{u}_{ki}, \bar{\delta}_{ki}} V_k \quad (196)$$

$$- \alpha_{k+li} \Delta \mathbf{u}_i^{max} \leq \Delta \mathbf{u}_i(k+l) \leq \alpha_{k+li} \Delta \mathbf{u}_i^{max} \quad l = 0, \dots, m_i - 1 \quad (197)$$

$$\Delta \mathbf{u}_i(k+l) = \mathbf{0} \quad l \geq m_i \quad (198)$$

$$\mathbf{u}_i^{min} \leq \mathbf{u}_i(k-1) + \sum_{q=0}^l \Delta \mathbf{u}_i(k+q) \leq \mathbf{u}_i^{max} \quad l = 0, \dots, m_i - 1 \quad (199)$$

$$\mathbf{x}_q^s(k) - \mathbf{y}_{kq}^{sp} - \boldsymbol{\delta}_{kqi} + \tilde{\mathbf{D}}_{qi}^0 \Delta \mathbf{u}_{ki} + \sum_{\substack{j=1 \\ j \neq i}}^M \left\{ \tilde{\mathbf{D}}_{qj}^0 \Delta \mathbf{u}_{kj} - \boldsymbol{\delta}_{kqj} \right\} = \mathbf{0} \quad q = 1, \dots, M \quad (200)$$

Aqui utilizou-se a mesma notação do Capítulo 4. Note somente que, como o horizonte de controle pode ser definido diferentemente de um subsistema para outro, $\Delta \mathbf{u}_{kj} = [\Delta \mathbf{u}_j^T(k|k) \Delta \mathbf{u}_j^T(k+1|k) \cdots \Delta \mathbf{u}_j^T(k+m_j-1|k)]^T$, $j = 1, \dots, M$.

Note também que a $\alpha_{kj} = 0$ garante que $\Delta \mathbf{u}_j(k|k) = \mathbf{0}$ para qualquer j por conta da restrição (197). Note também que a necessidade de se atualizar as variáveis de folga vem da restrição (200).

Assim como nos controladores anteriores, a implementação é feita de maneira paralela. É necessário portanto, um processo iterativo. O Algoritmo 1 do Capítulo 4, subseção 4.1.1, pode ser utilizado aqui, de modo que as questões de convergência, factibilidade recursiva e estabilidade são as mesmas já discutidas (seção 4.2).

Finalmente, cabe notar que, se $\alpha_{ki} = 0$ para um determinado subsistema i qualquer, o controlador poderia ser programado para não minimizar V_k naquele instante e assumir diretamente a solução obtida no instante anterior deslocada no tempo $\Delta \tilde{\mathbf{u}}_{k+1i} = [\Delta \mathbf{u}_i^{*T}(k|k-1) \cdots \Delta \mathbf{u}_i^{*T}(k+m_i-2|k-1) \mathbf{0}^T]^T$, $\tilde{\boldsymbol{\delta}}_{kji} = \boldsymbol{\delta}_{k-1ji}^*$, $j = 1, \dots, M$ reduzindo assim a carga computacional. Neste caso, as variáveis de folga cruzadas para tal sistema não seriam recalculadas e a função objetivo não seria monotonicamente decrescente. As ações de controle, entretanto, seriam as mesmas, pois elas são restritas a zero pela equação (197). Mais ainda, se somente um subsistema tem $\alpha_{ki} = 1$ em um determinado instante k , o Algoritmo 1 não precisa iterar, já que o único controlador a operar naquele instante o faz como se fosse um controlador centralizado. Com isso, o custo computacional é reduzido ainda mais, mas não se pode garantir uma função objetivo monotonicamente decrescente.

7.3 Resultados de simulação para sistemas simplificados com múltiplas escalas temporais

Para avaliar o desempenho dos controladores com múltiplas escalas temporais, utilizou-se um sistema 3×3 descrito pelas seguintes FTs:

$$\mathbf{G}(s) = \begin{bmatrix} \frac{5}{10s+1} & \frac{8}{8s+1} & 0 \\ 0 & \frac{3.3}{15s+1} & \frac{7}{10s+1} \\ \frac{6}{45s+1} & 0 & \frac{20}{50s+1} \end{bmatrix} \quad (201)$$

Observe que este sistema tem funções de transferência com dinâmicas mais lentas para as variáveis que se relacionam a y_3 . Assim, esta foi utilizada para ser a parte lenta do sistema. Com relação aos parâmetros, os seguintes foram considerados:

$$\begin{aligned} \Delta T &= 1\text{min e } 5\text{min} & p &= 2 \\ \mathbf{y}_0 &= [1 \quad 1 \quad 1]^T & m &= 6 \\ \mathbf{y}_k^{sp} &= [2 \quad 2 \quad 2]^T & \mathbf{Q} &= \text{diag}(1 \quad 1 \quad 1) \\ \mathbf{u}^{max} &= [0.5 \quad 0.5 \quad 0.1]^T & \mathbf{R} &= \text{diag}(1 \quad 1 \quad 1) \\ \mathbf{u}^{min} &= -\mathbf{u}^{max} & \mathbf{S} &= 10^3 \text{diag}(1 \quad 1 \quad 1) \\ \Delta \mathbf{u}^{max} &= [0.05 \quad 0.05 \quad 0.01]^T \end{aligned}$$

O modelo é nominal e o controlador utiliza Filtro de Kalman conforme descrito na seção 3.3. Quanto ao método iterativo, foi utilizado o Algoritmo 1 desenvolvido na subseção 4.1.1. Note que, para propósitos de comparação de simulação, todos os sub-sistemas têm o mesmo horizonte de controle. A extensão para horizontes distintos é considerada, entretanto, trivial.

Neste caso, é possível dividir este sistema em dois subsistemas arranjando (y_1, y_2, u_1, u_2) e (y_3, u_3) ou em três subsistemas: (y_1, u_1) , (y_2, u_2) e (y_3, u_3) , como já foi feito em casos anteriores. Os pesos foram distribuídos igualmente, ou seja: $\mathbf{w} = [0.5 \quad 0.5]^T$ e $\mathbf{w} = [1/3 \quad 1/3 \quad 1/3]^T$

As Figuras 16 e 17 fazem uma comparação entre três controladores: centralizado, distribuído e distribuído com diferentes escalas temporais. Os casos distribuídos são aqueles divididos em dois subsistemas no arranjo descrito acima. Já as Figuras 18 e 19 fazem uma comparação entre os mesmos controladores divididos em três subsistemas no arranjo também descrito acima.

Figura 16: Simulação: dois subsistemas com múltiplas escalas temporais

Fonte: elaboração própria (2020).

Figura 17: Função objetivo: dois subsistemas com múltiplas escalas temporais

Fonte: elaboração própria (2020).

Com relação aos tempos de computação, o controlador centralizado levou 13.69ms por período de simulação, o controlador distribuído sem diferentes escalas temporais levou 34.84ms por período de simulação e o controlador com diferentes escalas temporais levou 22.60ms. Como se pode observar, o tempo de computação diminuiu, pois a cada 5 períodos de simulação, em quatro deles o valor $\Delta \mathbf{u}_i = \mathbf{0}$ já é dado.

Figura 18: Simulação: três subsistemas com múltiplas escalas temporais

Fonte: elaboração própria (2020).

Figura 19: Função objetivo: três subsistemas com múltiplas escalas temporais

Fonte: elaboração própria (2020).

Para o caso em que o controlador é dividido em três subsistemas, os tempos de computação foram 25.52ms para o caso sem diferentes escalas temporais e 16.95ms para o caso com diferentes escalas temporais. Mais uma vez, uma maior subdivisão em diversos sistemas leva a menores tempos de computação, pois divide-se o controle em mais servidores.

8 Resultados de simulação para um caso industrial

Casos industriais normalmente têm uma quantidade muito maior de variáveis, dinâmicas mais complexas e variação de escalas muito maior que os casos apresentados até o momento. Neste ponto do trabalho, coube apresentar o desempenho do sistema desenvolvido para um caso industrial. Um bom exemplo de aplicação prática pode ser extraído de Martin, Zanin e Odloak [19]. Neste artigo, os autores avaliaram o comportamento de um controlador centralizado de horizonte infinito ligado a uma camada superior de otimização conforme apresentado por Alvarez e Odloak [28] para o caso do sistema de destilação de óleo cru da unidade de refino de Capuava-SP, pertencente à Petrobras, genericamente chamado de CDU. Para controlar tal sistema, o controlador descrito no Capítulo 5 é o mais adequado e é o que foi utilizado (os controladores dos Capítulos 6 e 7 não foram utilizados neste exemplo, dado que não há tempo morto, nem foram consideradas diferentes escalas de tempo).

O sistema, que pode ser observado na Figura 20, é composto das dessalgadoras O-527/502, por onde passa inicialmente o óleo cru, da coluna de pré-flash N-507, que separa a nafta leve e tem uma retirada lateral que pode ser adicionada ao pool de diesel, da fornalha L-506, da fracionadora principal N-506 com diversas retiradas laterais além de diversos trocadores de calor. Além disso, o pré-flash pode receber uma corrente de stripping com vapor super-aquecido.

Figura 20: Esquema da unidade de destilação de óleo cru da unidade de refino da Petrobras de Capuava-SP

Fonte: adaptado de Martin, Zanin e Odloak (2018) [19].

Na Figura 20, observa-se que há dez variáveis de interesse a serem controladas e oito a serem manipuladas, uma quantidade significativamente maior do que vinha sendo tratada nos casos simplificados. Estas, por sua vez, são descritas nas Tabelas 4 e 5

Tabela 4: Definição e valores mínimos, máximos e iniciais para as saídas - CDU

Variável	Definição	Unidade	y^{min}	y^{max}	y_0
y_1	N-507 massa de vapor por volume de cru	kg/m ³	5,6	6	5,8
y_2	Fluxo de nafta pesada	m ³ /d	300	304	302
y_3	N-506 refluxo	m ³ /d	1095	1125	1104
y_4	Temperatura de retirada de querosene	°C	181	184	182
y_5	Diesel ASTM D-86 95%	°C	368	372	370,9
y_6	Flash-point do diesel	°C	32	35	33,8
y_7	Ponto final nafta leve ASTM D-86	°C	174	175,5	175
y_8	N-507 refluxo	m ³ /d	1406	1413	1409
y_9	Fluxo de nafta leve	m ³ /d	1266	1280	1273
y_{10}	L-506 Carga térmica	Gcal/h	15.6	15.8	15.7

Fonte: adaptado de Martin, Zanin e Odloak (2018) [19].

Obs. Valores mínimos e máximos na Tabela 4 foram alterados em relação ao artigo original para a simulação. As faixas originais eram bem mais largas. Para os valores originais, vide Martin, Zanin e Odloak [19].

Tabela 5: Definição e valores mínimos e máximos para as entradas - CDU

Variável	Definição	Unidade	u^{min}	u^{max}	Δu^{max}
u_1	Fluxo de cru	m ³ /d	9299	9300	20.0
u_2	N-507 temperatura de topo	°C	126	128	0.1
u_3	N-507 stripping	t/h	1.2	1.8	0.03
u_4	N-506 temperatura de topo	°C	118	119	0.05
u_5	Bombeamento de diesel pesado	m ³ /d	550	2000	10.0
u_6	Refluxo de diesel	m ³ /d	3800	6500	8.0
u_7	Fluxo de querosene	m ³ /d	900	1000	6.0
u_8	L-506 temperatura de saída	°C	363	367	0.07

Fonte: adaptado de Martin, Zanin e Odloak (2018) [19].

O sistema descrito acima, modelo nominal, esquematizado na Figura 20 e detalhado nas Tabelas 4 e 5, pode ser representado pela seguinte dinâmica:

$$\mathbf{G}(s) = \begin{bmatrix} 0 & 0 & G_{1,3}(s) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{2,4}(s) & 0 & 0 & G_{2,7}(s) & 0 \\ 0 & G_{3,2}(s) & 0 & G_{3,4}(s) & 0 & G_{3,6}(s) & G_{3,7}(s) & G_{3,8}(s) \\ 0 & 0 & 0 & G_{4,4}(s) & 0 & G_{4,6}(s) & G_{4,7}(s) & G_{4,8}(s) \\ 0 & 0 & 0 & G_{5,4}(s) & G_{5,5}(s) & 0 & G_{5,7}(s) & G_{5,8}(s) \\ 0 & 0 & 0 & G_{6,4}(s) & 0 & G_{6,6}(s) & G_{6,7}(s) & 0 \\ 0 & G_{7,2}(s) & G_{7,3}(s) & 0 & 0 & 0 & 0 & 0 \\ G_{8,1}(s) & G_{8,2}(s) & G_{8,3}(s) & 0 & 0 & 0 & 0 & 0 \\ G_{9,1}(s) & G_{9,2}(s) & G_{9,3}(s) & 0 & 0 & 0 & 0 & 0 \\ G_{10,1}(s) & G_{10,2}(s) & G_{10,3}(s) & 0 & 0 & 0 & 0 & G_{10,8}(s) \end{bmatrix} \quad (202)$$

As equações específicas de $G_{i,j}(s)$ encontram-se detalhadas no Apêndice C. Olhando para esta matriz, é notável que u_1 , u_2 e u_3 têm pouca influencia em y_2 , y_3 , y_4 , y_5 e y_6 ao passo que u_4 , u_5 , u_6 , u_7 e u_8 têm pouca influência em y_1 , y_7 , y_8 , y_9 e y_{10} . Assim, a seguinte divisão de subsistemas (baseada em uma divisão por equipamentos) é proposta na Figura 21, ficando cada coluna e a fornalha em um subsistema. A mesma reflete as relações entrada/saída que contêm todas as restrições e recomendações mencionadas acima.

Figura 21: CDU e sua divisão em três subsistemas

Fonte: adaptado de Martin, Zanin e Odloak (2018) [19].

Em termos de variáveis, os três subsistemas podem ser escritos como: $(y_1, y_7, y_8, y_9, u_1, u_2, u_3)$, (y_{10}, u_8) e $(y_2, y_3, y_4, y_5, y_6, u_4, u_5, u_6, u_7)$. Foram feitas simulações com o IHMPC centralizado e com o DC-IHMPC com RTO. Foram considerados os seguintes parâmetros de sintonia:

$$\begin{aligned}
\Delta T &= 1\text{min} & \mathbf{R} &= \text{diag}(5 \quad 1 \quad 6.6 \quad 1 \quad 0.1 \quad 6 \quad 3 \quad 142) \\
m &= 2 & \mathbf{Q}_y &= \text{diag}(2 \quad 1 \quad 3 \quad 50 \quad 5 \quad 5 \quad 5 \quad 7 \quad 10 \quad 20) \\
p &= 2 \text{ e } 3 & \mathbf{Q}_u &= \text{diag}(1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1) \\
\mathbf{w} &= [1/3 \quad 1/3 \quad 1/3]^T & \mathbf{S}_u &= 10^6 \text{diag}(1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1) \\
& & \mathbf{S}_y &= 10^9 \text{diag}(1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)
\end{aligned}$$

Quanto às condições iniciais e os valores desejados, as Tabelas 6 e 7 sumarizam a situação das entradas.

Tabela 6: Valores iniciais e desejados para as entradas - CDU

Variável	u_0	u^{des}
u_1	9300	9300
u_2	128	127,9
u_3	1,8	1,8
u_4	119	119
u_5	1340	1380
u_6	4860	4825
u_7	997	1000
u_8	363	363

Fonte: elaboração própria (2020).

Para testar a mudança de zonas para as saídas após $t = 30$ min foi alterada a faixa de y_{10} e os valores máximo e mínimo para u_1 , conforme a Tabela 7.

Tabela 7: Novos valores de zona para as saídas, restrições para as entradas e alteração de alvo para as entradas - CDU

Variável	Mínimo	Máximo	Desejado
y_{10}	15,5	15,7	-
u_1	9270	9300	9270

Fonte: elaboração própria (2020).

As Figuras 22 a 38 ilustram os resultados da simulação para a unidade descrita acima considerando o sistema centralizado vs. o sistema distribuído subdividido em três sub-sistemas conforme descrito nos parágrafos anteriores. As Figuras 22 a 27 ilustram o

comportamento das entradas do sistema (os gráficos para u_4 e u_8 foram intencionalmente omitidos, pelo fato de suas variações na simulação serem irrelevantes), ao passo que as Figuras 28 a 37 ilustram o comportamento das saídas do sistema. Para os primeiros 30 minutos de simulação (em que os alvos são alcançáveis), é possível observar que o controlador centralizado é um pouco mais veloz do que o controlador distribuído para alcançar os alvos. Além disso, a condição $p = 3$ obtém melhores resultados do que $p = 2$, como esperado. Após 30 minutos, quando o alvo de u_1 e a zona de controle de y_{10} não são mais alcançáveis (conforme Tabela 7), o desempenho dos controladores não é significativamente diferente. Os alvos u_1 e u_2 não podem ser atingidos, pois y_{10} atinge o limite superior estabelecido. O estado estacionário para o qual o sistema é levado neste caso é o mesmo para o controlador centralizado e distribuído. Finalmente a Figura 38 ilustra o decrescimento da função objetivo. Nos primeiros 30 minutos, tal função tende a zero, já que os alvos são alcançáveis, na sequência, a mesma se estabiliza em um valor diferente de zero.

Figura 22: Simulação: caso industrial (CDU) - entrada u_1 (Fluxo de cru)

Fonte: elaboração própria (2020).

Figura 23: Simulação: caso industrial (CDU) - entrada u_2 (N-507 temperatura de topo)

Fonte: elaboração própria (2020).

Figura 24: Simulação: caso industrial (CDU) - entrada u_3 (N-507 stripping)

Fonte: elaboração própria (2020).

Figura 25: Simulação: caso industrial (CDU) - entrada u_5 (Bombeamento de diesel pesado)

Fonte: elaboração própria (2020).

Figura 26: Simulação: caso industrial (CDU) - entrada u_6 (Refluxo de diesel)

Fonte: elaboração própria (2020).

Figura 27: Simulação: caso industrial (CDU) - entrada u_7 (Fluxo de querosene)

Fonte: elaboração própria (2020).

Figura 28: Simulação: caso industrial (CDU) - saída y_1 (N-507 massa de vapor por volume de cru)

Fonte: elaboração própria (2020).

Figura 29: Simulação: caso industrial (CDU) - saída y_2 (Fluxo de nafta pesada)

Fonte: elaboração própria (2020).

Figura 30: Simulação: caso industrial (CDU) - saída y_3 (N-506 refluxo)

Fonte: elaboração própria (2020).

Figura 31: Simulação: caso industrial (CDU) - saída y_4 (Temperatura de retirada de querosene)

Fonte: elaboração própria (2020).

Figura 32: Simulação: caso industrial (CDU) - saída y_5 (Diesel ASTM D-86 95%)

Fonte: elaboração própria (2020).

Figura 33: Simulação: caso industrial (CDU) - saída y_6 (Flash-point do diesel)

Fonte: elaboração própria (2020).

Figura 34: Simulação: caso industrial (CDU) - saída y_7 (Ponto final nafta leve ASTM D-86)

Fonte: elaboração própria (2020).

Figura 35: Simulação: caso industrial (CDU) - saída y_8 (N-507 refluxo)

Fonte: elaboração própria (2020).

Figura 36: Simulação: caso industrial (CDU) - saída y_9 (Fluxo de nafta leve)

Fonte: elaboração própria (2020).

Figura 37: Simulação: caso industrial (CDU) - saída y_{10} (L-506 Carga térmica)

Fonte: elaboração própria (2020).

Figura 38: Função objetivo: caso industrial (CDU)

Fonte: elaboração própria (2020).

Este exemplo mostra que o DC-IHMPC é adequado para controlar sistemas industriais com camadas de otimização (RTO). Com relação ao tempo de computação, considerando

$m = 2$, o controlador centralizado levou 42.09ms em média a cada instante para computar a solução, já o controlador distribuído levou 25.30ms em média a cada instante com $p = 2$ e 39.5ms em média a cada instante com $p = 3$. Observe que nas simulações de casos simples, o sistema distribuído não conseguia ser mais veloz do que o centralizado enquanto que para o caso industrial (e mais complexo), a situação se inverte.

9 Considerações finais

9.1 Conclusões

No presente trabalho, foram apresentadas diversas arquiteturas para controladores distribuídos. Todas elas utilizaram como modelos de estado adaptações do *Output Prediction Oriented Model* (OPOM) para sistemas distribuídos, bem como todas utilizaram horizonte infinito de predição, que apresenta vantagens em termos de estabilidade. A estabilidade e a factibilidade recursiva destes controladores foi demonstrada, de modo que tais controladores estariam aptos a controlar uma planta industrial. Mais ainda, foi proposto um controlador capaz de lidar com as variáveis em múltiplas escalas de tempo, o que não é prático para controladores centralizados.

Simulações para casos simples em termos de escala e complexidade das Funções de Transferência foram apresentadas, de modo a ilustrar de maneira didática o comportamento do controlador. Além disso, foi mostrada uma simulação para um caso industrial: *Crude Distillation Unit* (CDU) pertencente a uma refinaria de petróleo da Petrobras - unidade de Capuava-SP. Todas as simulações se comportaram bem, mostrando que os controladores desenvolvidos são eficientes e, mesmo em operação sub ótima, são capazes de controlar uma planta industrial.

Desta forma, conclui-se este trabalho tendo os objetivos iniciais atingidos e uma contribuição para o universo dos MPCs realizada. Esta tecnologia, que vem sendo desenvolvida desde a década de 1970, mostra ainda apresentar diversas possibilidades de aprimoramento. O desenvolvimento da mesma contribui para a eficiência da indústria, em particular a indústria química, tendo portanto a prerrogativa de ajudar a diminuir desperdícios, economizar recursos e reduzir riscos ambientais. Fica assim clara a necessidade de tal desenvolvimento.

9.2 Sugestões para trabalhos futuros

Em termos de pesquisa para a tecnologia de controladores distribuídos, sugere-se estudos para desenvolvimento de controladores capazes de lidar com polos integradores. Desenvolvimentos nesta área para controladores centralizados já foram feitos. Normalmente, tais problemas tentam zerar as contribuições destes polos para a função custo simultaneamente à redução do custo para os outros polos, ou então em duas etapas, zerando os polos instáveis primeiramente, e admitindo que o controlador passa a se comportar a partir daí como um controlador sem os mesmos. Assim, não deve ser difícil fazer a adaptação destes controladores para o caso distribuído.

Ainda em termos de desenvolvimento tecnológico, neste trabalho admitiu-se que o modelo que representava a planta era preciso, de modo que o controlador apresenta estabilidade nominal (que desconsidera imprecisões no modelo). Uma possível extensão deste trabalho é a estabilidade robusta, em que admite-se a possibilidade de erros na obtenção do modelo. Tais erros costumam ser compensados por controladores com este tipo de estabilidade de duas formas: admitindo-se que o modelo correto é um dentre diversas possibilidades, ou então admitindo-se que o modelo correto é uma combinação convexa de alguns modelos propostos. Estes modelos propostos passam a atuar como uma base e, em suma, o modelo correto passa a ser uma combinação linear desta base dentro do espaço de modelos possíveis.

Finalmente, o DC-IHMPC apresenta, graças à descentralização, bastante flexibilidade e redução do tempo computacional (como o presente trabalho mostrou). Sugere-se assim a integração com outras funções pertinentes ao controle de uma planta, como inteligência artificial, entre outras e até a eventual criação de um software com estas funções embarcadas.

Referências bibliográficas

- [1] RAWLINGS, J. B.; STEWART, B. T. **Coordinating multiple optimization-based controllers: New opportunities and challenges.** *Journal of Process Control*, v. 18, p. 839–845, 2008.
- [2] FARINA, M.; SCATTOLINI, R. **Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems.** *Automatica*, v. 48, p. 1088–1096, 2012.
- [3] SANTANA, D. D.; MARTINS, A. M. F.; ODLOAK, D. **An efficient cooperative-distributed model predictive controller with stability and feasibility guarantees for constrained linear systems.** *Systems and Control Letters*, v. 41, p. 104701, 2020.
- [4] RICHALET, J. et al. **Algorithmic control of industrial processes.** *Proceedings of the 4th IFAC symposium on identification and system parameter estimation*, p. 1119–1176, 1976.
- [5] RICHALET, J. et al. **Model Predictive Heuristic Control: Applications to Industrial Processes.** *Automatica*, v. 14, n. 5, p. 413–428, 1978.
- [6] CUTLER, C. R.; RAMAKER, B. L. **Dynamic matrix control—a computer control.** *American Institute of Chemical Engineers national meeting*, Paper 51-C, Houton, TX, 1979.
- [7] CUTLER, C. R.; RAMAKER, B. L. **Dynamic matrix control—a computer control algorithm.** *Proceedings of the joint automatic control conference*, San Francisco, CA., 1980.
- [8] PRETT, D. M.; GILLETTE, R. D. **Optimization and constrained multivariable control of a catalytic cracking unit.** *Proceedings of the joint automatic control conference*. San Francisco, CA, Paper WP5-C, 1980.
- [9] MORSHEDI, A. M.; CUTLER, C. R.; SKROVANEK, T. A. **Optimal solution of dynamic matrix control with linear programming techniques (LDMC).** *American Control Conference*, p. 199–208, 1985.
- [10] GARCIA, C. E.; MORSHEDI, A. M. **Quadratic programming solution of dynamic matrix control (QDMC).** *Chemical Engineering Communications*, v. 46, p. 73–87, 1986.

- [11] MARQUIS, P.; BROUSTAIL, J. P. **SMOC, a bridge between state space and model predictive controllers: Application to the automation of a hydro-treating unit.** *IFAC Proceedings Volumes*, v. 21, n. 4, p. 37–45, 1988.
- [12] MORARI, M.; LEE, J. H. **Model predictive control: past, present and future.** *Computers and Chemical Engineering*, v. 23, n. 4-5, p. 667–682, 1999.
- [13] MICHALSKA, H.; MAYNE, D. Q. **Robust receding horizon control of constrained nonlinear systems.** *IEEE Transactions on Automatic Control*, v. 38, n. 11, p. 1623–1633, 1993.
- [14] INC, Honeywell. **RMPCT concepts reference.** Product literature from Honeywell Inc, 1995.
- [15] MUSKE, K. R.; RAWLINGS, J. B. **Model predictive control with linear models.** *American Institute of Chemical Engineers Journal*, v. 39, n. 2, p. 262–287, 1993.
- [16] MUSKE, K. R.; RAWLINGS, J. B. **The stability of constrained receding horizon control.** *IEEE Transactions on Automatic Control*, v. 38, n. 10, p. 1512–1516, 1993.
- [17] LEE, J. H.; MORARI, M.; GARCIA, C. E. **State-space interpretation of model predictive control.** *Automatica*, v. 30, n. 4, p. 707–717, 1994.
- [18] BADGWELL, T. A. **Robust model predictive control of stable linear systems.** *International Journal of Control*, v. 68, n. 4, p. 797–818, 1997.
- [19] 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, n. 3, p. 1205–1222, 2019.
- [20] ODLOAK, D. **Extended robust model predictive control.** *American Institute of Chemical Engineers Journal*, v. 50, n. 8, p. 1824–1836, 2004.
- [21] 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.
- [22] GONZÁLEZ, A. H.; SOTOMAYOR, O. A. Z.; ODLOAK, D. **Stable IHMPC for Unstable Systems.** *IFAC Proceedings of the 17th World Congress*, p. 6950–6955, 2008.
- [23] 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.

- [24] CARRAPIÇO, O. L.; ODLOAK, D. **A stable model predictive control for integrating processes.** *Computes and Chemical Engineering*, v. 29, p. 1089–1099, 2005.
- [25] GONZÁLEZ, A. H.; ODLOAK, D. **A stable MPC with zone control.** *Journal of Process Control*, v. 19, 2009.
- [26] GONZÁLEZ, A.H.; ODLOAK, D. **Robust Control, Theory and Applications.** A. Bartoszewicz. Cap. Robust model predictive control for time delayed systems with optimizing targets and zone control, p. 339–370, 2011.
- [27] GONZÁLEZ, A. H.; MARCHETTI, J. L.; D., Odloak. **Robust model predictive control with zone control.** *IET Control Theory and Applications*, v. 3, n. 1, p. 121–135, 2009.
- [28] ALVAREZ, L. A.; ODLOAK, D. **Robust integration of real time optimization with linear model predictive control.** *Computers and Chemical Engineering*, v. 34, n. 12, p. 1937–1944, 2010.
- [29] CAPRON, B. D. O.; ODLOAK, D. **A robust LQR-MPC control strategy with input constraints and control zones.** *Journal of Process Control*, v. 64, p. 89–99, 2018.
- [30] STEWART, B. T. et al. **Cooperative distributed model predictive control.** *Systems and Control Letters*, v. 59, p. 460–469, 2010.
- [31] VENKAT, A. N. et al. **Distributed MPC Strategies With Application to Power System Automatic Generation Control.** *IEEE Ttransactions on Control Systems Technology*, v. 16, n. 6, p. 1192–1206, 2008.
- [32] SCATTOLINI, R. **Architectures for distributed and hierarchical Model Predictive Control – A review.** *Journal of Process Control*, v. 19, p. 723–731, 2009.
- [33] FARINA, M. et al. **An Approach to Distributed Predictive Control for Tracking–Theory and Applications.** *IEEE Transactions on Control Systems Technology*, v. 22, n. 4, p. 1558–1566, 2014.
- [34] TRODDEN, P. A.; MONASTERIOS, P. R. B.; MAESTRE, J. M. **Distributed MPC with Minimization of Mutual Disturbance Sets.** *American Control Conference*, p. 5193–5198, 2016.
- [35] MENDES, P. R. C. et al. **A practical approach for hybrid distributed MPC.** *Journal of Process Control*, v. 55, p. 30–41, 2017.
- [36] LIU, J. et al. **Sequential and Iterative Architectures for Distributed Model Predictive Control of Nonlinear Process Systems.** *American Institute of Chemical Engineers Journal*, v. 56, n. 8, p. 2137–2149, 2010.

- [37] AL-GHERWI, W.; BUDMAN, H.; ELKAMEL, A. **A robust distributed model predictive control algorithm**. *Journal of Process Control*, v. 21, p. 1127–1137, 2011.
- [38] OGATA, Katsuhiko. **Modern Control Engineering**. Prentice Hall. Cap. Introduction to Control Systems, p. 13–15, 2009.
- [39] OGATA, Katsuhiko. **Modern Control Engineering**. Prentice Hall. Cap. Mathematical Modeling of Control Systems, p. 29–39, 2009.
- [40] SANTORO, B. F. «Controle preditivo de horizonte infinito para sistemas integradores e com tempo morto». Tese de mestrado. São Paulo: Escola Politécnica da Universidade de São Paulo. 2011.
- [41] ODLOAK, D. **Notes in "PQI-5780 - Advanced process control - model predictive control"**. 2018.
- [42] HESPANHA, J. P. **Linear Systems Theory**. 2^a ed. New Jersey: Princeton University Press. 2009.
- [43] OGATA, Katsuhiko. **Modern Control Engineering**. Prentice Hall. Cap. Mathematical Modeling of Control Systems, p. 675–687, 2009.
- [44] SIMON, D. **Optimal State Estimation**. John Wiley e Sons. 2006.
- [45] GLAD, T.; LJUNG, L. **Control Theory**. Taylor e Francis. 2000.
- [46] SIMON, D. **Optimal State Estimation**. John Wiley e Sons. 2006.
- [47] PATWARDHAN, S. C. et al. **Nonlinear Bayesian state estimation: A review of recent developments**. *Control Engineering Practice*, v. 20, p. 933–953, 2012.
- [48] SCHNEIDER, R.; GEORGAKIS, C. **How to not make the extended Kalman Filter fail**. *Industrial Engineering Chemistry Research*, v. 52, p. 3354–3362, 2013.
- [49] HASELTINE, E. L.; RAWLINGS, J. B. **Critical evaluation of extended Kalman Filtering and Moving-Horizon Estimation**. *Industrial Engineering Chemistry Research*, v. 44, p. 2451–2460, 2005.
- [50] BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. **Nonlinear Programming: theory and Algorithms**. 3^a ed. Wiley. 2006.
- [51] KASSMANN, D. E.; BADGWELL, T.; HAWKINGS, R. B. **Robust target calculation for model predictive control**. *American Institute of Chemical Engineers Journal*, v. 45, n. 5, p. 1007–1023, 2000.
- [52] SKOGESTAD, S. **Control structure design for complete chemical plants**. *Computers and Chemical Engineering*, v. 28, p. 219–234, 2004.

- [53] LONG, Y. et al. **A Hierarchical Distributed MPC for HVAC Systems.** *American Control Conference*, p. 2385–2390, 2016.
- [54] LI, H.; SWARTZ, C. L. E. **Approximation techniques for dynamic real-time optimization (DRTO) of distributed MPC systems.** *Computers and Chemical Engineering*, v. 118, p. 195–209, 2018.
- [55] MACIEJOWSKI, J. M. **Predictive Control with Constraints.** Prentice Hall. 2002.
- [56] OSBORNE, M. J.; RUBINSTEIN, A. **A Course in Game Theory.** MIT Press. 1994.
- [57] BAŞAR, T.; OLSDER, G. J. **Dynamic Noncooperative Game Theory.** SIAM. 1999.
- [58] BERTSEKAS, D. P. **Nonlinear Programming.** Athena Scientific. 1999.
- [59] XIA, C.; LIU, H. **Bi-level model predictive control for optimal coordination of multi-area automatic generation control units under wind power integration.** *Processes*, v. 7, n. 10, p. 669, 2019.
- [60] KONG, X. et al. **Hierarchical Distributed Model Predictive Control of Standalone Wind/Solar/Battery Power System.** *IEEE Transactions on Systems, Man and Cybernetics: Systems*, v. 49, n. 8, p. 1570–1581, 2019.
- [61] BALDEA, M.; DAOUTIDIS, P. **Multi-Scale Dynamics of High Energy Throughput Systems.** *IFAC Proceedings Volumes*, v. 40, n. 5, p. 203–208, 2007.
- [62] BALDEA, M.; DAOUTIDIS, P.; NAGY, Z. K. **Nonlinear Model Predictive Control of Integrated Process Systems.** *IFAC Proceedings Volumes*, v. 43, n. 15, p. 1040–1045, 2010.

Apêndice A - Função objetivo para o controlador definido no Capítulo 4 em sua forma matricial

Sejam $\mathbf{H}$, $\mathbf{C}_f$ e c as matrizes para os termos quadrático e linear e o termo constante da função objetivo, assim, pode-se reescrever (119) para um subsistema i qualquer:

$$V_k = \left\| \begin{bmatrix} \Delta \mathbf{u}_{ki} \\ \bar{\delta}_{ki} \end{bmatrix} \right\|_{\mathbf{H}}^2 + \mathbf{C}_f^T \begin{bmatrix} \Delta \mathbf{u}_{ki} \\ \bar{\delta}_{ki} \end{bmatrix} + c \quad (203)$$

Ou então, separando-se os termos que se referem às variáveis $\Delta \mathbf{u}_{ki}$ dos termos que se referem às variáveis de folga:

$$V_k = \begin{bmatrix} \Delta \mathbf{u}_{ki}^T & \bar{\delta}_{ki}^T \end{bmatrix} \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{ki} \\ \bar{\delta}_{ki} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{f1}^T & \mathbf{C}_{f2}^T \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{ki} \\ \bar{\delta}_{ki} \end{bmatrix} + c \quad (204)$$

As subdivisões que compõem a matriz $\mathbf{H}$, para o subsistema i , podem ser escritas da seguinte forma:

$$\mathbf{H}_{11} = \sum_{j=1}^M w_j \left\{ \left\| \mathbf{D}_{mj}^0 + \Psi_{gj} \mathbf{F}_{uj} \right\|_{\mathbf{Q}_{gj}}^2 + \left\| \mathbf{F}_{uj} \right\|_{\bar{\mathbf{Q}}_{gj}}^2 \right\} + w_i \mathbf{R}_{gi} \quad (205)$$

$$\mathbf{H}_{21} = \mathbf{H}_{12}^T = \begin{bmatrix} -w_1 \bar{\mathbf{I}}_1^T \mathbf{Q}_{g1} \{ \mathbf{D}_{m1}^0 + \Psi_{g1} \mathbf{F}_{u1} \} \\ -w_2 \bar{\mathbf{I}}_2^T \mathbf{Q}_{g2} \{ \mathbf{D}_{m2}^0 + \Psi_{g2} \mathbf{F}_{u2} \} \\ \vdots \\ -w_M \bar{\mathbf{I}}_M^T \mathbf{Q}_{gM} \{ \mathbf{D}_{mM}^0 + \Psi_{gM} \mathbf{F}_{uM} \} \end{bmatrix} \quad (206)$$

$$\mathbf{H}_{22} = \text{diag} \left(w_1 \left\{ \left\| \bar{\mathbf{I}}_1 \right\|_{\mathbf{Q}_{g1}}^2 + \mathbf{S}_1 \right\} \cdots w_M \left\{ \left\| \bar{\mathbf{I}}_M \right\|_{\mathbf{Q}_{gM}}^2 + \mathbf{S}_M \right\} \right) \quad (207)$$

Já as subdivisões da matriz $\mathbf{C}_f$ são dadas por:

$$\begin{aligned}
\mathbf{C}_{f1} = \sum_{j=1}^M w_j \left\{ \left[\mathbf{D}_{\mathbf{m}ji}^0 + \boldsymbol{\Psi}_{\mathbf{g}j} \mathbf{F}_{\mathbf{u}ji} \right]^T \mathbf{Q}_{gj} \left[\bar{\mathbf{I}}_j \left(\mathbf{x}_j^s(k) - \mathbf{y}_{kj}^{sp} - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{kjq} \right) + \right. \right. \\
\left. \left. \boldsymbol{\Psi}_{\mathbf{g}j} \mathbf{F}_{\mathbf{x}j} \mathbf{x}_j^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M [(\mathbf{D}_{\mathbf{m}jq}^0 + \boldsymbol{\Psi}_{\mathbf{g}j} \mathbf{F}_{\mathbf{u}jq}) \Delta \mathbf{u}_{kq}] \right] + \right. \\
\left. \mathbf{F}_{\mathbf{u}ji}^T \bar{\mathbf{Q}}_{gj} \left[\mathbf{F}_{\mathbf{x}j} \mathbf{x}_j^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M \mathbf{F}_{\mathbf{u}jq} \Delta \mathbf{u}_{kq} \right] \right\}
\end{aligned} \tag{208}$$

$$\mathbf{C}_{f2} = \begin{bmatrix} w_1 \bar{\mathbf{I}}_1^T \mathbf{Q}_{g1} \left[\bar{\mathbf{I}}_1 \left(\mathbf{x}_1^s(k) - \mathbf{y}_{k1}^{sp} - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{k1q} \right) + \right. \\ \left. \boldsymbol{\Psi}_{\mathbf{g}1} \mathbf{F}_{\mathbf{x}1} \mathbf{x}_1^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M [(\mathbf{D}_{\mathbf{m}1q}^0 + \boldsymbol{\Psi}_{\mathbf{g}1} \mathbf{F}_{\mathbf{u}1q}) \Delta \mathbf{u}_{kq}] \right] \\ \vdots \\ w_M \bar{\mathbf{I}}_M^T \mathbf{Q}_{gM} \left[\bar{\mathbf{I}}_M \left(\mathbf{x}_M^s(k) - \mathbf{y}_{kM}^{sp} - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{kMq} \right) + \right. \\ \left. \boldsymbol{\Psi}_{\mathbf{g}M} \mathbf{F}_{\mathbf{x}M} \mathbf{x}_M^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M [(\mathbf{D}_{\mathbf{m}Mq}^0 + \boldsymbol{\Psi}_{\mathbf{g}M} \mathbf{F}_{\mathbf{u}Mq}) \Delta \mathbf{u}_{kq}] \right] \end{bmatrix} \tag{209}$$

Finalmente, para o termo c , para o subsistema i , escreve-se:

$$\begin{aligned}
c = \sum_{j=1}^M w_j & \left\{ \left\| \bar{\mathbf{I}}_j \left(\mathbf{x}_j^s(k) - \mathbf{y}_{kj}^{sp} - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{kj q} \right) + \boldsymbol{\Psi}_{\mathbf{g}j} \mathbf{F}_{\mathbf{x}j} \mathbf{x}_j^d(k) \right. \right. \\
& + \sum_{\substack{q=1 \\ q \neq i}}^M \left[(\mathbf{D}_{\mathbf{m}jq}^0 + \boldsymbol{\Psi}_{\mathbf{g}j} \mathbf{F}_{\mathbf{u}jq}) \Delta \mathbf{u}_{kq} \right] \left\|_{\mathbf{Q}_{gj}}^2 + \left\| \mathbf{F}_{\mathbf{x}j} \mathbf{x}_j^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M \mathbf{F}_{\mathbf{u}jq} \Delta \mathbf{u}_{kq} \right\|_{\bar{\mathbf{Q}}_{gj}}^2 \right. \\
& \left. \left. + \|\Delta \mathbf{u}_{kj \neq i}\|_{\mathbf{R}_{gj}}^2 + \sum_{\substack{q=1 \\ q \neq i}}^M \|\boldsymbol{\delta}_{kj q}\|_{\mathbf{S}_j}^2 \right\} \right.
\end{aligned} \tag{210}$$

Apêndice B - Função objetivo para o controlador definido no Capítulo 5 em sua forma matricial

Assim como no Apêndice A, sejam $\mathbf{H}$, $\mathbf{C}_f$ e c as matrizes para os termos quadrático e linear e o termo constante da função objetivo, assim, pode-se reescrever (166) para um subsistema i qualquer:

$$V_k = \left\| \begin{bmatrix} \Delta \mathbf{u}_{ki} \\ \mathbf{y}_{ki}^{sp} \\ \bar{\delta}_{ki} \\ \delta_{ki}^u \end{bmatrix} \right\|_{\mathbf{H}}^2 + \mathbf{C}_f^T \begin{bmatrix} \Delta \mathbf{u}_{ki} \\ \mathbf{y}_{ki}^{sp} \\ \bar{\delta}_{ki} \\ \delta_{ki}^u \end{bmatrix} + c \quad (211)$$

Novamente, separando-se os termos que se referem às variáveis $\Delta \mathbf{u}_{ki}$ e $\mathbf{y}_{ki}^{sp}$ dos termos que se referem às variáveis de folga:

$$V_k = \begin{bmatrix} \Delta \mathbf{u}_{ki} & \mathbf{y}_{ki}^{sp} & \bar{\delta}_{ki} & \delta_{ki}^u \end{bmatrix} \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} & \mathbf{H}_{14} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} & \mathbf{H}_{24} \\ \mathbf{H}_{31} & \mathbf{H}_{32} & \mathbf{H}_{33} & \mathbf{H}_{34} \\ \mathbf{H}_{41} & \mathbf{H}_{42} & \mathbf{H}_{43} & \mathbf{H}_{44} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{ki} \\ \mathbf{y}_{ki}^{sp} \\ \bar{\delta}_{ki} \\ \delta_{ki}^u \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{f1}^T & \mathbf{C}_{f2}^T & \mathbf{C}_{f3}^T & \mathbf{C}_{f4}^T \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{ki} \\ \mathbf{y}_{ki}^{sp} \\ \bar{\delta}_{ki} \\ \delta_{ki}^u \end{bmatrix} + c \quad (212)$$

As subdivisões que compõem a matriz $\mathbf{H}$, para o subsistema i , podem ser escritas da seguinte forma:

$$\mathbf{H}_{11} = \sum_{j=1}^M w_j \left\{ \left\| \mathbf{D}_{\mathbf{m}ji}^0 + \Psi_{\mathbf{g}j} \mathbf{F}_{\mathbf{u}ji} \right\|_{\mathbf{Q}_{\mathbf{g}j}}^2 + \left\| \mathbf{F}_{\mathbf{u}ji} \right\|_{\bar{\mathbf{Q}}_{\mathbf{g}j}}^2 \right\} + w_i \left(\left\| \mathbf{D}_{\mathbf{m}i}^u \right\|_{\mathbf{Q}_{\mathbf{g}i}^u}^2 + \mathbf{R}_{\mathbf{g}i} \right) \quad (213)$$

$$\mathbf{H}_{21} = \mathbf{H}_{12}^T = -w_i \bar{\mathbf{I}}_i^T \mathbf{Q}_{\mathbf{g}i} \left\{ \mathbf{D}_{\mathbf{m}ii}^0 + \Psi_{\mathbf{g}i} \mathbf{F}_{\mathbf{u}ii} \right\} \quad (214)$$

$$\mathbf{H}_{31} = \mathbf{H}_{13}^T = \begin{bmatrix} -w_1 \bar{\mathbf{I}}_1^T \mathbf{Q}_{\mathbf{g}1} \{ \mathbf{D}_{\mathbf{m}1i}^0 + \Psi_{\mathbf{g}1} \mathbf{F}_{\mathbf{u}1i} \} \\ -w_2 \bar{\mathbf{I}}_2^T \mathbf{Q}_{\mathbf{g}2} \{ \mathbf{D}_{\mathbf{m}2i}^0 + \Psi_{\mathbf{g}2} \mathbf{F}_{\mathbf{u}2i} \} \\ \vdots \\ -w_M \bar{\mathbf{I}}_M^T \mathbf{Q}_{\mathbf{g}M} \{ \mathbf{D}_{\mathbf{m}Mi}^0 + \Psi_{\mathbf{g}M} \mathbf{F}_{\mathbf{u}Mi} \} \end{bmatrix} \quad (215)$$

$$\mathbf{H}_{41} = \mathbf{H}_{14}^T = w_i \bar{\mathbf{I}}_{nu_i}^T \mathbf{Q}_{\mathbf{g}i}^u \mathbf{D}_{\mathbf{m}i}^u \quad (216)$$

$$\mathbf{H}_{22} = w_i \left\| \bar{\mathbf{I}}_i \right\|_{\mathbf{Q}_{\mathbf{g}i}}^2 \quad (217)$$

$$\mathbf{H}_{32} = \mathbf{H}_{23}^T = \begin{bmatrix} \mathbf{0}_{(i-1) \times nu_i} \\ w_i \left\| \bar{\mathbf{I}}_i \right\|_{\mathbf{Q}_{\mathbf{g}i}}^2 \\ \mathbf{0}_{(M-i) \times nu_i} \end{bmatrix} \quad (218)$$

$$\mathbf{H}_{42} = \mathbf{H}_{24}^T = \mathbf{0} \quad (219)$$

$$\mathbf{H}_{33} = \text{diag} \left(w_1 \left\{ \left\| \bar{\mathbf{I}}_1 \right\|_{\mathbf{Q}_{\mathbf{g}1}}^2 + \mathbf{S}_1^y \right\} \cdots w_M \left\{ \left\| \bar{\mathbf{I}}_M \right\|_{\mathbf{Q}_{\mathbf{g}M}}^2 + \mathbf{S}_M^y \right\} \right) \quad (220)$$

$$\mathbf{H}_{43} = \mathbf{H}_{34}^T = \mathbf{0} \quad (221)$$

$$\mathbf{H}_{44} = w_i \left\{ \left\| \bar{\mathbf{I}}_i \right\|_{\mathbf{Q}_{\mathbf{g}i}^u}^2 + \mathbf{S}_i^u \right\} \quad (222)$$

Já as subdivisões da matriz $\mathbf{C}_f$ são dadas por:

$$\begin{aligned}
\mathbf{C}_{f1} = & \sum_{j=1}^M w_j \left\{ \left[\mathbf{D}_{\mathbf{m}ji}^0 + \boldsymbol{\Psi}_{\mathbf{g}j} \mathbf{F}_{\mathbf{u}ji} \right]^T \mathbf{Q}_{gj} \left[\bar{\mathbf{I}}_j \left(\mathbf{x}_j^s(k) - \mathbf{y}_{kj \neq i}^{sp} - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{kj q} \right) + \right. \right. \\
& \left. \left. \boldsymbol{\Psi}_{\mathbf{g}j} \mathbf{F}_{\mathbf{x}j} \mathbf{x}_j^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M \left[(\mathbf{D}_{\mathbf{m}jq}^0 + \boldsymbol{\Psi}_{\mathbf{g}j} \mathbf{F}_{\mathbf{u}jq}) \Delta \mathbf{u}_{kq} \right] \right] + \right. \\
& \left. \mathbf{F}_{\mathbf{u}ji}^T \bar{\mathbf{Q}}_{gj} \left[\mathbf{F}_{\mathbf{x}j} \mathbf{x}_j^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M \mathbf{F}_{\mathbf{u}jq} \Delta \mathbf{u}_{kq} \right] \right\} + \\
& w_i [\mathbf{D}_{\mathbf{m}i}^u]^T \mathbf{Q}_{\mathbf{g}i}^u \bar{\mathbf{I}}_{nu_i} (\mathbf{u}_i(k-1) - \mathbf{u}_{ki}^{des})
\end{aligned} \tag{223}$$

$$\begin{aligned}
\mathbf{C}_{f2} = & w_i \bar{\mathbf{I}}_i^T \mathbf{Q}_{\mathbf{g}i} \left[\bar{\mathbf{I}}_i \left(\mathbf{x}_i^s(k) - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{ki q} \right) + \right. \\
& \left. \boldsymbol{\Psi}_{\mathbf{g}i} \mathbf{F}_{\mathbf{x}i} \mathbf{x}_i^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M \left[(\mathbf{D}_{\mathbf{m}iq}^0 + \boldsymbol{\Psi}_{\mathbf{g}i} \mathbf{F}_{\mathbf{u}iq}) \Delta \mathbf{u}_{kq} \right] \right]
\end{aligned} \tag{224}$$

$$\begin{aligned}
\mathbf{C}_{f3} = & \begin{bmatrix} w_1 \bar{\mathbf{I}}_1^T \mathbf{Q}_{g1} \left[\bar{\mathbf{I}}_1 \left(\mathbf{x}_1^s(k) - \mathbf{y}_{k1 \neq i}^{sp} - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{k1 q} \right) + \right. \\ \left. \boldsymbol{\Psi}_{\mathbf{g}1} \mathbf{F}_{\mathbf{x}1} \mathbf{x}_1^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M \left[(\mathbf{D}_{\mathbf{m}1q}^0 + \boldsymbol{\Psi}_{\mathbf{g}1} \mathbf{F}_{\mathbf{u}1q}) \Delta \mathbf{u}_{kq} \right] \right] \\ \vdots \\ w_M \bar{\mathbf{I}}_M^T \mathbf{Q}_{gM} \left[\bar{\mathbf{I}}_M \left(\mathbf{x}_M^s(k) - \mathbf{y}_{kM \neq i}^{sp} - \sum_{\substack{q=1 \\ q \neq i}}^M \boldsymbol{\delta}_{kM q} \right) + \right. \\ \left. \boldsymbol{\Psi}_{\mathbf{g}M} \mathbf{F}_{\mathbf{x}M} \mathbf{x}_M^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M \left[(\mathbf{D}_{\mathbf{m}Mq}^0 + \boldsymbol{\Psi}_{\mathbf{g}M} \mathbf{F}_{\mathbf{u}Mq}) \Delta \mathbf{u}_{kq} \right] \right] \end{bmatrix}
\end{aligned} \tag{225}$$

$$\mathbf{C}_{f4} = w_i \bar{\mathbf{I}}_{nu_i}^T \mathbf{Q}_{\mathbf{g}i}^u \bar{\mathbf{I}}_{nu_i} (\mathbf{u}_i(k-1) - \mathbf{u}_{ki}^{des}) \quad (226)$$

Finalmente, para o termo c , para o subsistema i , escreve-se:

$$\begin{aligned} c = & \sum_{j=1}^M w_j \left\{ \left\| \bar{\mathbf{I}}_j \left(\mathbf{x}_j^s(k) - \mathbf{y}_{kj \neq i}^{sp} - \sum_{\substack{q=1 \\ q \neq i}}^M \delta_{kj q} \right) \Psi_{\mathbf{g}j} \mathbf{F}_{\mathbf{x}j} \mathbf{x}_j^d(k) + \right. \right. \\ & + \sum_{\substack{q=1 \\ q \neq i}}^M [(\mathbf{D}_{\mathbf{m}jq}^0 + \Psi_{\mathbf{g}j} \mathbf{F}_{\mathbf{u}jq}) \Delta \mathbf{u}_{kq}] \left. \right\|_{\mathbf{Q}_{\mathbf{g}j}}^2 + \left\| \mathbf{F}_{\mathbf{x}j} \mathbf{x}_j^d(k) + \sum_{\substack{q=1 \\ q \neq i}}^M \mathbf{F}_{\mathbf{u}jq} \Delta \mathbf{u}_{kq} \right\|_{\bar{\mathbf{Q}}_{\mathbf{g}j}}^2 \\ & + \left\| \Delta \mathbf{u}_{kj \neq i} \right\|_{\mathbf{R}_{\mathbf{g}j}}^2 + \sum_{\substack{q=1 \\ q \neq i}}^M \left\| \delta_{kj q} \right\|_{\mathbf{S}_j}^2 \left. \right\} \\ & + \sum_{\substack{q=1 \\ q \neq i}}^M \left\{ \left\| \bar{\mathbf{I}}_{nu_q} (\mathbf{u}_q(k-1) - \mathbf{u}_{kq}^{des}) \right\|_{\mathbf{Q}_{\mathbf{g}q}^u}^2 + \left\| \delta_{kq}^u \right\|_{\mathbf{S}_q^u}^2 \right\} \end{aligned} \quad (227)$$

Apêndice C - Funções de transferência (CDU)

$$G_{1,3} = \frac{0.46s^3 - 1.15s^2 + 7.4s + 0.91}{s^4 + 2.57s^3 + 11.9s^2 + 3.96s + 0.32}$$

$$G_{2,4} = \frac{0.24s^6 - 0.14s^5 + 1.71s^4 + 3.63s^3 + 3.72s^2 + 0.66s + 0.05}{s^7 + 5.58s^6 + 19.65s^5 + 39.65s^4 + 11.93s^3 + 3.85s^2 + 0.2s + 4.13 \times 10^{-3}}$$

$$G_{2,7} = \frac{0.01s^5 - 0.02s^4 - 0.02s^3 - 0.01s^2 - 1.84 \times 10^{-4}s - 1.89 \times 10^{-4}}{s^6 + 3.39s^5 + 4.97s^4 + 2.45s^3 + 0.47s^2 + 0.03s + 9.58 \times 10^{-4}}$$

$$G_{3,2} = \frac{-1.29s^2 - 1.37s - 2.13}{s^3 + 1.99s^2 + 0.99s + 0.19}$$

$$G_{3,4} = \frac{0.14s^5 + 3.82s^4 - 7.56s^3 - 12.02s^2 - 0.3s - 0.39}{s^6 + 5.22s^5 + 18.97s^4 + 17.63s^3 + 3.14s^2 + 0.32s + 0.01}$$

$$G_{3,6} = \frac{-0.05s^3 + 0.52s^2 - 1.92s - 2.32}{s^4 + 4.56s^3 + 17.19s^2 + 17.01s + 5.58}$$

$$G_{3,7} = \frac{-1.84 \times 10^{-3}s^2 - 6.35 \times 10^{-4}s - 1.07 \times 10^{-4}}{s^3 + 0.26s^2 + 0.02s + 3.83 \times 10^{-4}}$$

$$G_{3,8} = \frac{0.5407s^6 + 1.312s^5 + 0.5546s^4 + 0.3327s^3 + 0.06388s^2 + 0.005414s + 0.3131 \times 10^{-3}}{s^7 + 0.4485s^6 + 0.599s^5 + 0.1516s^4 + 0.06437s^3 + 0.009542s^2 + 0.5524 \times 10^{-3}s + 1.289 \times 10^{-5}}$$

$$G_{4,4} = \frac{-0.13s^2 + 0.26s + 0.29}{s^3 + 4.16s^2 + 1.9s + 0.32}$$

$$G_{4,6} = \frac{0.01s^3 + 0.03s^2 + 0.05s - 0.44}{s^4 + 7.28s^3 + 29.63s^2 + 53.21s + 33.54}$$

$$G_{4,7} = \frac{-6.67 \times 10^{-4}s^4 + 1.26 \times 10^{-3}s^3 + 1.71 \times 10^{-3}s^2 + 1.68 \times 10^{-4}s + 6.58 \times 10^{-5}}{s^5 + 2.48s^4 + 3.34s^3 + 0.66s^2 + 0.08s + 4.09 \times 10^{-3}}$$

$$G_{4,8} = \frac{0.008947s^3 - 0.3574s^2 + 1.39s + 0.9825}{s^4 + 4.132s^3 + 15.16s^2 + 7.629s + 1.396}$$

$$G_{5,4} = \frac{-6.42 \times 10^{-4}s^6 - 0.02s^5 + 0.04s^4 + 0.05s^3 + 0.04s^2 + 0.03s + 6.327 \times 10^{-3}}{s^7 + 4.27s^6 + 16.04s^5 + 10.95s^4 + 6.98s^3 + 1.72s^2 + .51s + 0.02}$$

$$G_{5,5} = \frac{-6.499 \times 10^{-4}s^2 - 6.382 \times 10^{-4}s - 1.923 \times 10^{-4}}{s^3 + 0.58s^2 + 0.09s + 5.371 \times 10^{-3}}$$

$$G_{5,7} = \frac{-8.51 \times 10^{-4}s^5 - 1.52 \times 10^{-4}s^4 - 1.48 \times 10^{-4}s^3 - 1.01 \times 10^{-5}s^2 - 1.32 \times 10^{-6}s - 6 \times 10^{-8}}{s^6 + 0.31s^5 + 0.21s^4 + 0.04s^3 + 6.14 \times 10^{-3}s^2 + 2.93 \times 10^{-4}s + 5.26 \times 10^{-6}}$$

$$G_{5,8} = \frac{-0.02s^5 - 0.26s^4 + 0.74s^3 + 0.21s^2 + 0.11s + 0.04}{s^6 + 5.85s^5 + 20.46s^4 + 13.84s^3 + 4.03s^2 + 0.67s + 0.04}$$

$$G_{6,4} = \frac{0.03s^3 - 0.12s^2 + 0.6s + 0.42}{s^4 + 3.26s^3 + 13.41s^2 + 7.94s + 1.64}$$

$$G_{6,6} = \frac{9.46 \times 10^{-4}s^2 - 1.39 \times 10^{-5}s + 1.72 \times 10^{-4}}{s^3 + 0.46s^2 + 0.12s + 0.01}$$

$$G_{6,7} = \frac{-8.431 \times 10^{-4}s^3 + 5.02 \times 10^{-5}s^2 - 0.02s - 3.67 \times 10^{-3}}{s^4 + 3.34s^3 + 13.05s^2 + 3.09s + 0.25}$$

$$G_{7,2} = \frac{4.6 \times 10^{-3}s^2 + 0.06s + 4.11 \times 10^{-3}}{s^3 + 0.39s^2 + 0.07s + 2.93 \times 10^{-3}}$$

$$G_{7,3} = \frac{-0.2s^3 - 1.08s^2 + 3.67s + 3.13}{s^4 + 9.85s^3 + 35.88s^2 + 12.34s + 1.62}$$

$$G_{8,1} = \frac{7.31 \times 10^{-4}s^5 - 1.82 \times 10^{-3}s^4 + 1.49 \times 10^{-2}s^3 + 2.67 \times 10^{-3}s^2 + 1.32 \times 10^{-3}s + 1.71 \times 10^{-4}}{s^6 + 2.18s^5 + 5.83s^4 + 2.14s^3 + 0.56s^2 + 0.03s + 9.07 \times 10^{-4}}$$

$$G_{8,2} = \frac{11.31s^6 + 102.9s^5 - 58.29s^4 - 171.2s^3 - 362.1s^2 - 159.7s - 56.27}{s^7 + 8.84s^6 + 38.89s^5 + 71.07s^4 + 83.25s^3 + 53.24s^2 + 19.11s + 3.16}$$

$$G_{8,3} = \frac{-0.83s^2 + 15.19s + 4.86}{s^3 + 1.85s^2 + 0.46s + 0.06}$$

$$G_{9,1} = \frac{-2.63 \times 10^{-2}s^6 - 0.17s^5 + 0.36s^4 + 0.15s^3 + 0.12s^2 + 5.55 \times 10^{-3}s + 8.86 \times 10^{-4}}{s^7 + 8.16s^6 + 29.45s^5 + 19.87s^4 + 7.98s^3 + 1.11s^2 + 0.1s + 3.2 \times 10^{-3}}$$

$$G_{9,2} = \frac{-8.8s^5 + 13.69s^4 + 8.6s^3 + 5.95s^2 + 0.89s + 0.18}{s^6 + 4.13s^5 + 4.94s^4 + 2.19s^3 + 0.65s^2 + 0.11s + 7.57 \times 10^{-3}}$$

$$G_{9,3} = \frac{0.56s^2 + 14.74s + 5.08}{s^3 + 1.96s^2 + 0.52s + 0.05}$$

$$G_{10,1} = \frac{-1.88 \times 10^{-4}s^5 + 1.11 \times 10^{-4}s^4 + 1.53 \times 10^{-4}s^3 + 8.67 \times 10^{-5}s^2 - 1.76 \times 10^{-5}s + 3.9 \times 10^{-5}}{s^6 + 3.83s^5 + 6.67s^4 + 2.84s^3 + .68s^2 + .12s + 4.88 \times 10^{-3}}$$

$$G_{10,2} = \frac{-4.67 \times 10^{-3}s^2 - 5.59 \times 10^{-4}s - 6.76 \times 10^{-4}}{s^3 + 0.32s^2 + 0.06s + 5.19 \times 10^{-3}}$$

$$G_{10,3} = \frac{1.82 \times 10^{-3}s^3 + 0.2s^2 - 0.41s - 0.78}{s^4 + 3.55s^3 + 14.1s^2 + 9.24s + 2.22}$$

$$G_{10,8} = \frac{0.03s^3 - 0.1s^2 + .51s + 0.11}{s^4 + 2.96s^3 + 12.66s^2 + 5.6s + 0.7}$$