



COPPE
UFRJ

AN INVESTIGATION OF RTO USING TRANSIENT MEASUREMENTS IN THE
ABSENCE OF A RIGOROUS DYNAMIC PROCESS MODEL

Pedro de Azevedo Delou

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Química, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Química.

Orientadores: Argimiro Resende Secchi
Maurício Bezerra de Souza
Júnior

Rio de Janeiro
Agosto de 2024

AN INVESTIGATION OF RTO USING TRANSIENT MEASUREMENTS IN THE
ABSENCE OF A RIGOROUS DYNAMIC PROCESS MODEL

Pedro de Azevedo Delou

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM
CIÊNCIAS EM ENGENHARIA QUÍMICA.

Examinada por:

Prof. Argimiro Resende Secchi, D.Sc.

Prof. Maurício Bezerra de Souza Júnior, D.Sc.

Eng. André Domingues Quelhas, D.Sc.

Prof. Galo Antonio Carrillo Le Roux, D.Sc.

Prof. Márcio André Fernandes Martins, D.Sc.

Prof. Príamo Albuquerque Melo Junior, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

AGOSTO DE 2024

Delou, Pedro de Azevedo

An investigation of RTO using transient measurements in the absence of a rigorous dynamic process model/Pedro de Azevedo Delou. – Rio de Janeiro: UFRJ/COPPE, 2024.

XXXIV, 220 p.: il.; 29, 7cm.

Orientadores: Argimiro Resende Secchi

Maurício Bezerra de Souza Júnior

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Química, 2024.

Referências Bibliográficas: p. 170 – 199.

1. Real-time optimization. 2. Model Predictive Control. 3. Modifier adaptation. 4. Gaussian Process. 5. Kalman Filter. I. Secchi, Argimiro Resende *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

*Aos meus sobrinhos e sobrinha: Tito, Nicolas, Lucas,
Oliver e Sofia.*

Agradecimentos

Aos professores Argimiro e Maurício, pela orientação incansável ao longo deste trabalho. Agradeço pela dedicação, incentivo e suporte contínuo, bem como pela confiança e pelas oportunidades concedidas no início da minha carreira, que abriram portas inimagináveis para mim. Vocês são exemplos de profissionalismo e inspiração para inúmeras gerações de engenheiros(as) químicos(as) formados(as) no Brasil.

Aos colaboradores desta tese que contribuíram como coautores dos artigos produzidos: Rodrigo Curvelo, Rafael Demuner, José Matias, Johannes Jäschke, Bruno Capron e Lucas Bernardino. Um agradecimento especial ao José Matias, que realizou o experimento no laboratório da NTNU, cujos resultados foram essenciais para esta tese. A colaboração de todos vocês foi fundamental.

Aos membros da banca examinadora, Márcio Martins, Galo Le Roux, André Quelhas e Príamo A. Melo Jr., pela atenção, disponibilidade e tempo dedicado à avaliação do meu trabalho, bem como pelas discussões enriquecedoras durante a arguição.

A todos os meus professores, desde a formação fundamental, passando pelo ensino médio, graduação, mestrado e doutorado. Cada um de vocês deixou uma marca que levarei comigo por toda a vida.

Aos meus antigos líderes na Siemens, Steve Hall e Graham Moffatt, por me concederem tempo de trabalho para a execução da minha tese de doutorado e por me incentivarem a continuar perseguindo meus objetivos acadêmicos.

À minha família, pelo apoio e suporte incondicionais ao longo da minha vida. Em especial aos meus queridos sobrinhos e sobrinha, Tito, Nicolas, Lucas, Oliver e Sofia. Vocês trouxeram alegria e enriqueceram minha vida com suas lembranças e sorrisos, mesmo à distância.

À minha esposa, Marina, pelo apoio, compreensão e incentivo durante os períodos em que precisei me ausentar para me dedicar a este trabalho. Sem você, nada disso seria possível. Te amo.

Aos novos amigos que fiz em Londres, que tornam a distância mais fácil de suportar: Lorena, Henrique, Leandro e Rozanna. Um carinho especial aos pequenos, Tiago e Isabella.

Aos amigos que ficaram no Brasil, mas que, de alguma forma, continuam próximos, mesmo que fisicamente distantes.

Aos amigos do LADES, do LMSCP e do PEQ, que trilharam caminhos semelhantes, enfrentando desafios únicos. A convivência e as discussões com todos foram de imenso valor.

À UFRJ, por ter me proporcionado toda a minha formação acadêmica de forma gratuita, especialmente à Escola de Química, onde me graduei, e ao Programa de Engenharia Química da COPPE, onde realizei meu mestrado e agora concluo meu doutorado.

Finalmente, às instituições de fomento à pesquisa no Brasil — FAPERJ, CNPq, Capes e Petrobras — por todas as bolsas e incentivos recebidos ao longo da minha formação acadêmica, mesmo que não tenha recebido financiamento durante o doutorado.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

UMA INVESTIGAÇÃO DE RTO COM USO DE MEDIDAS TRANSIENTES NA AUSÊNCIA DE UM MODELO DINÂMICO RIGOROSO DE PROCESSO

Pedro de Azevedo Delou

Agosto/2024

Orientadores: Argimiro Resende Secchi
Maurício Bezerra de Souza Júnior

Programa: Engenharia Química

A otimização em tempo real (RTO) sofre com longas esperas por estado estacionário. O RTO híbrido (HRTO) resolve esta desvantagem combinando uma estimação dinâmica com otimização estacionária. Este trabalho visa explorar novas formas de lidar com o HRTO na ausência de um modelo dinâmico rigoroso. Propusemos uma estrutura Hammerstein que combina o modelo estático e um modelo dinâmico linear identificado com dados da planta. Esta estrutura foi avaliada em diversas arquiteturas de HRTO. As arquiteturas com dinâmica fixa superaram as atualizadas devido ao surgimento de modos oscilatórios indesejáveis. Introduzimos um HRTO integrando objetivos econômicos em um MPC adaptativo de horizonte infinito usando princípios de controle auto-otimizante (SOC). A abordagem preservou a observabilidade e superou o RTO clássico com menor custo computacional. Abordamos a incerteza estrutural no modelo estático usando adaptação por modificadores (MA), processo gaussiano (GP) e conceitos de região de confiança. Uma MA de saída (MAy) baseado em GP foi proposto em uma estrutura HRTO empregando um NMPC Hammerstein. A abordagem apresentou desempenho semelhante ao MA-GP, com a flexibilidade adicional de utilizar os modificadores de saída em diferentes cálculos baseados em modelos, como no NMPC. Finalmente, avaliamos um controlador baseado em Aprendizagem por Reforço (RL) e o comparamos com um NMPC ideal. Ambos os controladores atingem os objetivos de controle com pequenas diferenças de desempenho. As diferentes alterações de HRTO ampliam as oportunidades para a sua implementação, mesmo em cenários sem um modelo dinâmico rigoroso.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

AN INVESTIGATION OF RTO USING TRANSIENT MEASUREMENTS IN THE
ABSENCE OF A RIGOROUS DYNAMIC PROCESS MODEL

Pedro de Azevedo Delou

August/2024

Advisors: Argimiro Resende Secchi
Maurício Bezerra de Souza Júnior

Department: Chemical Engineering

The classic Real-time optimization (RTO) suffers from long waits for steady operation. The hybrid RTO (HRTO) addresses this drawback by combining dynamic estimation with static economic optimization. This work aims to explore new ways to deal with HRTO in the absence of a rigorous dynamic process model. We proposed a Hammerstein structure combining the static model and a linear autoregressive exogenous (ARX) model identified from plant data. This structure is evaluated across various HRTO architectures, including uncertain parameters. The fixed dynamics architectures outperformed the dynamically updated ones because of the rise of undesirable oscillatory modes. We introduced an HRTO framework integrating economic objectives via self-optimizing control (SOC) principles in an adaptive infinite-horizon MPC. The approach preserved observability and outperformed the classic RTO with a lower computational cost. We addressed structural uncertainty in the static model using modifier adaptation (MA), Gaussian process (GP), and trust-region concepts. An output MA (MAy) based on GP was proposed in an HRTO framework employing a Hammerstein NMPC. The approach presented a similar performance to the MA-GP, with the added flexibility of using the output modifiers in different model-based layers. Finally, we assessed a Reinforcement Learning (RL) actor-critic-based controller and compared it with an ideal NMPC. Both controllers effectively achieved control objectives under unmeasured disturbances with minor performance differences. The different developed HRTO alternatives expand the opportunities for its implementation, including scenarios without rigorous dynamic models.

Contents

Agradecimientos	v
List of Figures	xiv
List of Tables	xix
List of Symbols	xx
List of Abbreviations	xxxii
1 Introduction	1
1.1 Objective	4
1.2 Thesis structure	5
1.3 Main scientific contributions of the thesis	5
2 Literature Review	8
2.1 Steady-state Optimization Formulation	8
2.2 Static RTO Methods Classification	11
2.3 The two-step approach	14
2.3.1 Model Adequacy	15
2.3.2 Stability Conditions	16
2.4 The ISOPE approaches	19
2.4.1 Original ISOPE methodology	19
2.4.2 ISOPE coping with Inequality Constraints	20
2.4.3 ISOPE with bias correction term	21
2.5 Modifier adaptation	21
2.5.1 Model Adequacy	23
2.5.2 Stability Conditions	24
2.5.3 MA variations	25
2.5.3.1 Constraint adaptation	25
2.5.3.2 MA with output modification	25
2.5.3.3 MA with Lagrangian modification	26

2.5.3.4	Directional MA	26
2.5.3.5	Second-order MA	28
2.5.3.6	Nested MA	29
2.5.3.7	Dual MA	29
2.5.3.8	MA with Trust-Region formulation	32
2.5.4	MA using Approximate Models	33
2.5.4.1	Convex approximations	33
2.5.4.2	Subscript to refer to quadratic approximations . . .	34
2.5.4.3	Gaussian Process	37
2.5.4.4	RBF Neural Network	42
2.6	Use of Transient Measurement	44
2.6.1	Parameter adaptation schemes	45
2.6.1.1	Seminal works	46
2.6.1.2	HRTO or ROPA	48
2.6.2	Problem adaptation schemes	52
2.6.2.1	Seminal works	52
2.6.2.2	Fast RTO or fast MA	53
2.6.2.3	MAWQA using transient measurements	57
2.6.3	Optimizing control approaches	58
2.6.3.1	Self-optimizing Control	59
2.6.3.2	Reinforcement Learning	64
3	Development of a hybrid real-time optimization approach in the absence of a rigorous dynamic model via process identification of Hammerstein model structures	67
3.1	Introduction	67
3.2	Hammerstein model structure	68
3.3	HRTO architectures	68
3.3.1	Hammerstein EKF algorithm	69
3.3.2	RELS algorithm	70
3.4	Case Study: The Williams-Otto Reactor	70
3.5	Conclusion	74
4	Steady-state real-time optimization using transient measurements in the absence of a dynamic mechanistic model: A framework of HRTO integrated with Adaptive Self-Optimizing IHMPC	76
4.1	Introduction	76
4.2	Hammerstein structure as an approximate dynamic model	79
4.2.1	Linearization of the Hammerstein structure and State-Space formulation	81

4.3	Real-time Adaptive Self-Optimizing Control Framework	82
4.3.1	Parameter Estimation based on transient measurements	84
4.3.2	Adaptive Self-Optimizing IHMPC based on the Hammerstein structure	85
4.3.3	Steady-state Economic Optimization	87
4.3.4	Obtaining Self-Optimizing variables and handling changes in Active Constraints	88
4.4	Case Study: The Williams-Otto Reactor	92
4.4.1	Hammerstein dynamic matrices identification	93
4.4.2	HEKF performance in open loop in the presence of measurement noise	95
4.4.3	Closed-loop performance of the Adaptive Self-Optimizing Control Framework	98
4.5	Conclusions	106
5	Steady-state real-time optimization using transient measurements and approximated Hammerstein dynamic model: A proof of concept in an experimental rig	108
5.1	Introduction	108
5.2	ROPA using Hammerstein dynamic model approximation	110
5.2.1	Hammerstein model structure	111
5.2.2	Identification of the dynamic matrix	113
5.2.3	Dynamic parameter estimation	114
5.2.4	Steady-state economic optimization	115
5.2.5	Algorithmic implementation	115
5.3	Case study: Experimental rig	117
5.3.1	Experiment layout	117
5.3.2	Process model	118
5.3.3	Optimization problem	120
5.3.4	Hammerstein model identification	120
5.3.5	ROPA closed-loop disturbance scenario	122
5.3.6	EKF tuning	124
5.4	Experimental results	125
5.4.1	Identification results	125
5.4.2	ROPA and discussion	126
5.5	Conclusion	132
6	Output Modifier adaptation based on Gaussian Process: Simultaneous use in Real-Time Optimization and Hammerstein NMPC	134
6.1	Introduction	134

6.2	Preliminaries	135
6.2.1	Problem Formulation	135
6.2.2	Modifier adaptation	136
6.2.3	Gaussian Process Approximation	137
6.3	Methodology	138
6.3.1	Trust-region MA-GP Scheme	138
6.3.2	Proposed Trust-region MAy-GP Scheme	139
6.3.3	Proposed NMPC with GP output modifiers formulation	139
6.4	Case Study: Williams-Otto Reactor	141
6.4.1	Comparison between MA-GP and MAy-GP	142
6.4.2	NMPC with GP output modifiers	144
6.5	Conclusion	146
7	A Comparison Between Process Control Strategies: Reinforcement Learning with RBFs and NMPC Coupled with EKF	148
7.1	Introduction	148
7.2	The RL-RBF controller	150
7.3	The NMPC-EKF controller	152
7.4	The Van de Vusse reactor Benchmark	153
7.4.1	The process	153
7.4.2	Definition of the control problem	155
7.5	Results	156
7.5.1	Training of the RL-RBF Controller	156
7.5.1.1	Effect of the number of training episodes	157
7.5.1.2	Effect of the training exploration	158
7.5.1.3	Effect of the concentration maximization activation function width used in the reward function	159
7.5.1.4	Effect of the weight on concentration maximization in the reward function	160
7.5.2	NMPC-EKF controller settings	161
7.5.3	Performance comparison between the NMPC-EKF and the RL-RBF controllers	162
7.6	Conclusion	163
8	Final Remarks	164
9	Future Work	167
	Bibliography	170
A	Formulation of the terminal weight of the Self-Optimizing IHMPC	200

B	Implementation of the Self-Optimizing IHMPC as a QP	202
C	Supplementary Material for Chapter 4: Closed-loop performance of the Adaptive SOC Framework in the presence of measurement noise	205
D	The Reinforcement Learning Problem	214

List of Figures

2.1	RTO classification flowcharts. HRTO: hybrid real-time optimization; ISOPE: integrated system optimization and parameter estimation; CA: constraint adaptation; MA: modifier adaptation; ESC: extremum-seeking control; NCO: Necessary condition of optimality; NEC: neighboring-extremal control; SOC: self-optimizing control.	11
2.2	Visual representation of the MA approach at k^{th} RTO iteration: (a) Modified cost function; (b) Modified generic constraint i .	23
2.3	Classic static RTO framework scheme.	44
2.4	Besl's RTO framework scheme (BESL <i>et al.</i> , 1998).	47
2.5	Prior and Lopez's RTO framework scheme (PRIOR and LOPEZ, 1999).	47
2.6	Valluru et al.'s RTO framework scheme (VALLURU <i>et al.</i> , 2015).	49
3.1	Performance of the different HRTO architecture over a scenario of parametric variation: (a) estimation of F_A with HEKF and RELS using model 1; (b) estimation of F_A with HEKF and RELS using model 2; (c) estimation of W with HEKF and RELS using model 1; (d) estimation of W with HEKF and RELS using model 2.	71
3.2	Performance of the different HRTO architecture over a scenario of parametric variation: (a) behavior of F_B with HEKF and RELS using model 1; (b) behavior of F_B with HEKF and RELS using model 2; (c) behavior of T_R with HEKF and RELS using model 1; (d) behavior of T_R with HEKF and RELS using model 2.	72
4.1	Block diagram of the proposed Hammerstein structure.	80
4.2	Proposed Real-time Adaptive Self-Optimizing Control Framework.	83
4.3	Input profile in the Hammerstein dynamic matrices identification: (a) F_B (kg/s); (b) T_R ($^{\circ}C$).	94
4.4	Output profile in the Hammerstein dynamic matrices identification: (a) w_E ; (b) w_P .	95

4.5	Disturbances estimation performance in open loop in the presence of measurement noise: (a) F_A (kg/s) estimation with tuning 1; (b) F_A (kg/s) estimation with tuning 2.	96
4.6	Output estimation performance with tuning 1 in open loop in the presence of measurement noise: (a) w_A ; (b) w_B ; (c) w_E ; (d) w_P . . .	97
4.7	Output estimation performance with tuning 2 in open loop in the presence of measurement noise: (a) w_A ; (b) w_B ; (c) w_E ; (d) w_P . . .	98
4.8	Closed-loop performance using Hammerstein model 1: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal inputs obtained by the HRTO.	100
4.9	Closed-loop performance using Hammerstein model 1: (a) w_E ; (b) w_P ; (c) Objective function, J_{eco} ($\$/s$); (d) Accumulated loss function ($\$$). Dashed lines represent quantities estimated by the HEKF, dotted lines represent the optimal setpoints obtained by the HRTO and the grey area represents the violation of the imposed constraint.	101
4.10	Closed-loop performance using Hammerstein model 2: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal setpoints obtained by the HRTO.	102
4.11	Closed-loop performance using Hammerstein model 2: (a) w_E ; (b) w_P ; (c) Objective function, J_{eco} ($\$/s$); (d) Accumulated loss function ($\$$). Dashed lines represent quantities estimated by the HEKF, dotted lines represent the optimal setpoints obtained by the HRTO and the grey area represents the violation of the imposed constraint.	103
4.12	Evolution of the slack variable using Hammerstein model 1 and tuning 1: (a) w_E with $w_{w_E}^U = 10$; (b) w_E with $w_{w_E}^U = 10^{-5}$; (c) $\rho_{w_E}^U$ with $w_{w_E}^U = 10$; (d) $\rho_{w_E}^U$ with $w_{w_E}^U = 10$	104
5.1	Block diagram comparing classic RTO, ROPA, and Hammerstein ROPA.	111
5.2	Experimental rig schematic adapted from MATIAS <i>et al.</i> (2022). . .	117
5.3	Identification scenario - input profile imposed to the experimental rig.	121
5.4	Identification scenario - output responses.	122
5.5	Reservoir valve opening profile. Different gray-shaded background highlights the five disturbance periods.	123
5.6	Identification result - liquid flow rate. Blue line denotes the response from Model 1, which contains the identified matrix A^h , and red line denotes the response from Model 2, which considers only the steady-state model.	125

5.7	Identification result - riser pressure. Blue line denotes the response from Model 1, which contains the identified matrix A^h , and red line denotes the response from Model 2, which considers only the steady-state model.	126
5.8	Parameter estimation - reservoir valve coefficients.	128
5.9	Parameter estimation - top valve coefficients.	129
5.10	Optimal input profile - gas injection flow rate.	130
5.11	Profit and average profit difference comparison between approaches.	131
6.1	20 RTO iterations for 30 noise realizations. The red dots are initial GP training points. Green triangles are the last iteration of each RTO run. Black lines are the constraint limits and the black star is the plant optimum.	143
6.2	Histogram of the last iteration point of the 30 noise realization runs. The solution frequency axis shows how many solutions of the 30-noise realization runs stayed between each objective band.	144
6.3	Dynamic evolution of the RTO objective function, and the manipulated variables, F_B and T_R	145
6.4	MAY-GP and Hammerstein NMPC iterations with trust regions evolution. The red dots are initial GP training points. Green triangles are the RTO solutions accepted by the trust-region algorithm and the black star is the plant optimum.	146
7.1	Open-loop non-minimal phase response of the Van de Vusse CSTR.	155
7.2	Influence of the number of episodes on training.	158
7.3	Influence of exploration on training.	159
7.4	Influence of the concentration maximization activation function width used in the reward function.	160
7.5	Effect of the weight on concentration maximization in the reward function.	161
7.6	Performance comparison between RL with RBF approximation and NMPC.	163
C.1	Closed-loop performance using Hammerstein model 1 in the presence of 1 % measurement noise: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal inputs obtained by the HRTO.	206

C.2	Closed-loop performance using Hammerstein model 1 in the presence of 1 % measurement noise: (a) w_G ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function (\$). Dots represent the noisy measurements, solid lines represent quantities estimated by the HEKF, dashed lines represent the optimal setpoints obtained by the HRTO, and the grey area represents the violation of the imposed constraint.	207
C.3	Closed-loop performance using Hammerstein model 1 in the presence of 5 % measurement noise: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal inputs obtained by the HRTO.	208
C.4	Closed-loop performance using Hammerstein model 1 in the presence of 5 % measurement noise: (a) w_G ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function (\$). Dots represent the noisy measurements, solid lines represent quantities estimated by the HEKF, dashed lines represent the optimal setpoints obtained by the HRTO, and the grey area represents the violation of the imposed constraint.	209
C.5	Closed-loop performance using Hammerstein model 2 in the presence of 1 % measurement noise: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal setpoints obtained by the HRTO.	210
C.6	Closed-loop performance using Hammerstein model 2 in the presence of 1 % measurement noise: (a) w_G ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function (\$). Dots represent the noisy measurements, solid lines represent quantities estimated by the HEKF, dashed lines represent the optimal setpoints obtained by the HRTO, and the grey area represents the violation of the imposed constraint.	211
C.7	Closed-loop performance using Hammerstein model 2 in the presence of 5 % measurement noise: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal inputs obtained by the HRTO.	212

C.8	Closed-loop performance using Hammerstein model 2 in the presence of 5 % measurement noise: (a) w_G ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function ($\$$). Dots represent the noisy measurements, solid lines represent quantities estimated by the HEKF, dashed lines represent the optimal setpoints obtained by the HRTO, and the grey area represents the violation of the imposed constraint.	213
D.1	The actor-critic framework (SUTTON and BARTO, 2018b). License: creativecommons.org/licenses/by-nc-nd/2.0/legalcode	218

List of Tables

3.1	nMSE comparison between HRT0 architectures	72
3.2	rMSE comparison between HRT0 architectures	73
3.3	Average consumption time for each stage in loop (<i>ms</i>)	73
3.4	Minimum consumption time for each stage in loop (<i>ms</i>)	74
3.5	Maximum consumption time for each stage in loop (<i>ms</i>)	74
4.1	Terminal accumulated terminal loss for each closed-loop approach	104
4.2	Performance of the proposed framework in terms of computational cost	105
5.1	Computational cost comparison between ROPA and HROPA	132
7.1	Van de Vusse reactor parameters	154
7.2	Initial conditions	155
7.3	Chosen centers for each state variable.	156
7.4	RL tuning parameters values.	156
7.5	Default training parameters for the sensitivity analysis	157
7.6	Training parameters for the RL-RBF controller	162

List of Symbols

$(\bar{\cdot})$	Notation to refer to the positional variable, p. 79
$(\cdot)^+$	Denotes <i>a posteriori</i> estimated quantities, p. 85
$(\cdot)^-$	Denotes <i>a priori</i> estimated quantities, p. 85
$(\cdot)^\dagger$	Moore-Penrose pseudo-inverse, p. 27
$(\cdot)^{dyn}$	Superscript to refer to the dynamic model, p. 55
$(\cdot)^h$	Superscript to refer to the Hammerstein model, p. 79
$(\cdot)^{lb}$	Superscript to denote lower bound, p. 20, 115
$(\cdot)^{sp}$	Superscript to refer to setpoints, p. 88
$(\cdot)^s$	Superscript to refer to the static model, p. 55
$(\cdot)^{ub}$	Superscript to denote upper bound, p. 20, 115
$(\cdot)_{nom}$	Subscript to refer to nominal values, p. 81
$(\cdot)_{qa}$	Subscript to refer to the quadratic approximation, p. 35
$(\dot{\cdot})$	Notation to denote estimated quantities, p. 85
F	Williams-Otto total mass flowrate, $F_A + F_B$, p. 70, 92, 141
F_i	Williams-Otto mass flowrate of $i \in [A, B]$, p. 70, 92, 141
J	Economic optimization objective function, p. 8
J_j^{diff}	Approach j profit percentage difference to ROPA, p. 130
J_c	Convex approximation of the cost function, p. 33
J_k	Objective function at the k^{th} run of the control problem, p. 86
J_{ec}	Economic objective function of the optimization problem, p. 115

$K_{i,j}$	Covariance matrix, p. 38, 137
L	Prediction horizon, p. 152
N	Control horizon, p. 86, 141, 152
$N(\mathbf{s})$	Number of times the closest center to \mathbf{s} has been visited, p. 151
N_{id}	Number of time samples in the identification set, p. 113
P_{pump}	Pump outlet pressure, p. 119
$P_{r,i}$	Pressure at the riser head of well i , p. 119
$P_{u,i}$	Pressure upstream to the injection point of well i , p. 119
$Q_{g,i}$	Gas flow rate for well i , p. 120
Q_{max}	Maximum gas throughput, p. 120
T_R	Williams-Otto reactor temperature, p. 70, 92, 141
W	Williams-Otto reactor mass, p. 70, 92
$\Delta \mathbf{u}$	Sequence of N control moves, p. 86, 152
$\Delta \mathbf{u}_{max}$	Upper limit for control moves, p. 152
Δ_k	Radius of the ball function $\mathcal{B}(\mathbf{u}_k, \Delta_k)$, p. 32
$\Delta_{f,k}$	Radius of the separated trust region for each Gaussian Process predictor, p. 40
Δ_{qa}	Ellipsoid scaling factor, p. 35
Γ	Matrix for the second dual constraint, p. 31
$\Gamma(\cdot)$	Nonlinear mapping representing the set of optimal solutions, p. 17
$\Gamma^n(\mathbf{u}_k)$	The n^{th} recursion of Γ starting from \mathbf{u}_k , p. 18
Λ	Scaling matrix, p. 137
$\Phi(\cdot)$	Cumulative normal distribution function, p. 41
Σ	Matrix for the first dual constraint, p. 31
α	Real number such that $0 < \alpha < 1$, p. 17

α_V	Learning rate for θ_V , p. 151
α_π	Learning rate for Θ_π , p. 151
$\bar{U}_k(\mathbf{u})$	Squared matrix at the k^{th} iteration, p. 31
β	Positive constant, p. 18
β_k	Sequence in the convergence condition, p. 33
\mathbf{A}	Augmented transition state in the incremental linear state space, p. 82
\mathbf{A}^h	Hammerstein dynamic matrix, p. 68, 79, 112, 140
\mathbf{B}^h	Hammerstein static matrix, p. 68, 79, 140
\mathbf{F}	Sensitivity of optimal measurements with respect to disturbances, p. 61, 89
\mathbf{F}_k	State transition matrix, p. 85
\mathbf{G}	Set of inequality constraints of the model's economic optimization problem, p. 9
\mathbf{G}^d	Gain matrix with respect to the disturbances, p. 81
\mathbf{G}^u	Gain matrix with respect to the inputs, p. 81
\mathbf{G}_c	Convex approximation of the constraints, p. 33
\mathbf{G}_p	Set of inequality constraints of the plant's economic optimization problem, p. 8
\mathbf{G}_u	Measurement Jacobian matrix with respect to \mathbf{u} , p. 60
\mathbf{G}_θ	Measurement Jacobian matrix with respect to θ , p. 60
\mathbf{H}	Self-optimizing output matrix, p. 59, 89
\mathbf{I}_{n_u}	Identity matrix of dimension $n_u \times n_u$, p. 27
\mathbf{J}_{ud}	Hessian matrix of the optimization cost function with respect to the disturbances, p. 89
\mathbf{J}_{uu}	Hessian matrix of the optimization cost function with respect to the input variables, p. 89
\mathbf{K}	Exponential filter matrix, p. 23

K_k	Kalman Filter gain, p. 85, 153
P	Terminal cost penalty matrix, p. 87, 141
P_k	Estimate covariance matrix, p. 85, 153
P_r	Measured riser head pressure vector, p. 119
Q_J	Cost function convex approximation quadratic term, p. 34
Q_k	State covariance matrix, p. 85, 153
Q_l	Measured liquid volumetric flow rate vector, p. 119
Q_θ	Covariance matrix of parameter noise, p. 69
Q_y	Covariance matrix of output noise, p. 69
R_k	Measurement covariance matrix, p. 85, 153
U^{id}	Set of input data of the identification problem, p. 113
U_k	Matrix of simplex directions, p. 29
U_r	Matrix of privileged input directions, p. 27
V	Identification output weight matrix, p. 14
W	Matrix adjusting the width of Gaussian bell-shaped functions, p. 150
W_c	Weight matrix for the self-optimizing variables, p. 86
W_d	Scaling matrix parameter disturbances, p. 89
W_u	Weight matrix for the inputs, p. 141
W_y	Weight matrix for the outputs, p. 141, 152
$W_{\Delta u}$	Weighting matrix for control moves, p. 86, 141, 152
W_θ	Scaling matrix for parameter disturbances, p. 61
W_{id}	Squared definite positive weight matrix for the identification problem., p. 114
W_{w_y}	Scaling matrix for measurement noise, p. 61, 89
Y	Matrix of disturbances and noise, p. 61, 89

Y^{id}	Set of measured output in the identification problem, p. 113
Z	Nulls-space matrix of the reduced Hessian of the plant's Lagrangian function, p. 9
Λ	Scaling matrix, p. 38
Λ_k	Collection of all modifiers, p. 24
$\Phi(\mathbf{s})$	Vector of n_c state-dependent Gaussian RBFs, p. 150
Ψ	Hyperparameters of Gaussian Process, p. 137
Σ_j	Covariance matrix of the estimated error at the j^{th} iteration, p. 56
Σ_{qa}	Covariance matrix of the regression set, p. 35
Θ^{id}	Set of uncertain parameters and disturbances in the identification problem, p. 113
Θ_k^J	Second-order modifier of the cost function, p. 28
$\Theta_k^{g_i}$	Second-order modifier of constraint i , p. 28
Θ_π	Learned parameters for the policy, p. 150
ϵ_k	Mismatch between the measurement and estimate of the plant output at time instant k , p. 152
$\hat{\theta}_k$	Set of optimal parameters at the k^{th} RTO run, p. 14
$\lambda(\cdot)$	Eigenvalues of the matrix (\cdot) , p. 18, 81, 113
λ^y	First-order modifier for the output variables, p. 26
λ_k	First-order modifier of the k^{th} RTO iteration, p. 20
$\lambda_k^{\mathcal{L}}$	First-order modifier for the Lagrangian function, p. 26
μ	Lagrangian multipliers vector of the model's economic optimization problem, p. 9
μ^*	Optimal Lagrangian multipliers vector of the model's economic optimization problem, p. 9
μ_p	Lagrangian multipliers vector of the plant's economic optimization problem, p. 9

$\boldsymbol{\mu}_p^*$	Optimal Lagrangian multipliers vector of the plant's economic optimization problem, p. 8
$\boldsymbol{\nu}$	Exploration noise vector, p. 151
$\boldsymbol{\rho}_u^L$	Lower slack variables for control variables, p. 87
$\boldsymbol{\rho}_u^U$	Upper slack variables for control variables, p. 87
$\boldsymbol{\rho}_y^L$	Lower slack variables for measured variables, p. 87
$\boldsymbol{\rho}_y^U$	Upper slack variables for measured variables, p. 87
$\boldsymbol{\theta}$	Model parameters, p. 9, 68, 111
$\boldsymbol{\theta}'$	Scaled deviation from nominal parameter conditions, p. 61
$\boldsymbol{\theta}_V$	Learned parameters for the value function, p. 150
$\boldsymbol{\theta}_{res}$	Reservoir valve flow coefficients vector, p. 119
$\boldsymbol{\theta}_{top}$	Top valve flow coefficients vector, p. 119
$\boldsymbol{\Gamma}$	Observation transition matrix, p. 85
$\boldsymbol{\varepsilon}^y$	Zeroth-order modifier for the output variables, p. 26
$\boldsymbol{\varepsilon}_k$	Zeroth-order modifier of the k^{th} RTO iteration, p. 21
$\boldsymbol{\varepsilon}_k^{\mathcal{L}}$	Zeroth-order modifier for the inequality constraint, p. 26
\boldsymbol{a}	Action vector of the RL-RBF controller, p. 150
\boldsymbol{a}_J	Cost function convex approximation linear term, p. 34
\boldsymbol{a}_{G_i}	Constraint i convex approximation linear term, p. 34
\boldsymbol{c}	Self-optimizing variables, p. 60
\boldsymbol{c}	Vector of n_c Gaussian RBF centers in the state space, p. 150
\boldsymbol{c}_s	Optimal setpoints for self-optimizing variables, p. 60
\boldsymbol{d}	Model parameters and unmeasured disturbances, p. 79
\boldsymbol{n}_k	Normal vector to the hyperplane \mathcal{H}_k , p. 30
\boldsymbol{p}	Liquid price vector, p. 120
$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{U})$	Vector of kernel evaluations, p. 137

s'	State vector after the implementation of action a , p. 151
s	State vector of the RL-RBF controller, p. 150
u	Optimization decision variables, p. 8, 68, 79, 135, 152
u^*	Optimal decision variables vector of the model's economic optimization problem, p. 9
u_m	Point in the metric space \mathbb{U} , p. 17
u_n	Point in the metric space \mathbb{U} , p. 17
u_p^*	Optimal decision variables vector of the plant's economic optimization problem, p. 8
u_∞	Unique fixed point of Γ in \mathbb{U} , p. 17
v^*	Point that minimizes the gradient error norm within the polyhedral set, p. 32
w'	Scaled deviation from nominal noise conditions, p. 61
w_y	Measurement noise vector, p. 60, 69
$w_{\rho_u^L}$	Weights vector for lower slack variables of control variables, p. 87
$w_{\rho_u^U}$	Weights vector for upper slack variables of control variables, p. 87
$w_{\rho_y^L}$	Weights vector for lower slack variables of measured variables, p. 87
$w_{\rho_y^U}$	Weights vector for upper slack variables of measured variables, p. 87
w_θ	Parameter noise vector, p. 69
x	State variables, p. 9, 68, 79, 111, 152
y	Predicted output variables, p. 9, 68, 79, 111, 152
y_p	Measured output variables, p. 8, 135
z	Short-term memory vector, p. 150
δ	Temporal difference error, p. 151

$\delta_{i,j}$	Kronecker delta function, p. 38, 137
δ_{noise}	Noise level, p. 31
ℓ_{min}	Nearest complement affine subspaces distance, p. 31
$\epsilon^f(\mathbf{u}, \mathbf{v})$	Gradient error due to truncation evaluated at point \mathbf{v} , p. 32
ϵ_{upper}	Desired upper bound of the gradient error norm, p. 31
η_1, η_2	Tuned scalar constants, p. 32
γ	Discount factor, p. 151
γ	Exploration weight, p. 41
γ_1, γ_2	Tuned scalar constants, p. 32
κ	Constant in the convergence condition, p. 33
$\kappa_k(\mathbf{u})$	Condition number at the k^{th} iteration, p. 31
$\ \cdot\ $	Norm, p. 33
λ_i	Hyperparameter controlling the length scale along each input dimension, p. 137
\mathbb{U}	Metric space, p. 17
\mathbb{U}_c	Set of all collected past operating points, p. 34
\mathbb{W}	Set of individual wells, p. 119
\mathbb{Y}_k^a	Set of all active constraints evaluated at time instant k , p. 90
\mathcal{A}_{EI}	Expected improvement acquisition function, p. 41
\mathcal{A}_{LCB}	Lower confidence bound acquisition function, p. 41
$\mathcal{B}_{ga}(\mathbf{u}_k, \Delta_{ga})$	Ellipsoid search space centered at \mathbf{u}_k with scaling factor Δ_{ga} , p. 35
$\mathcal{D}_k(\mathbf{u})$	General duality constraint to ensure regularity of U_{k+1} , p. 30
$\mathcal{D}_{1,k}(\mathbf{u})$	Dual constraint to keep small gradient variance, p. 31
$\mathcal{D}_{2,k}(\mathbf{u})$	Dual constraint to keep small step length between iterations, p. 31

$\mathcal{F}(\cdot)$	Steady state model states mapping function, p. 9, 68, 79, 111, 135
\mathcal{GP}	Gaussian Process, p. 38
$\mathcal{H}(\cdot)$	Steady state model output mapping function, p. 9, 68, 79, 111, 135
\mathcal{H}_k	Hyperplane defined by the matrix of simplex directions, p. 30
\mathcal{J}	Objective function of the model's economic optimization problem, p. 9, 135
\mathcal{J}_p	Objective function of the plant's economic optimization problem, p. 8
\mathcal{J}_{id}	Identification objective function, p. 14
\mathcal{L}	Lagrangian function of the model's economic optimization problem, p. 9
\mathcal{L}_p	Lagrangian function of the plant's economic optimization problem, p. 9
\mathcal{P}	Set of coefficients of the quadratic approximation function, p. 35
$\mathcal{U}_{d,k}$	Distant set, p. 34
$\mathcal{U}_{n,k}$	Neighborhood set, p. 34
$\mathcal{U}_{r,k}$	Set of points selected for regression at the k^{th} RTO iteration, p. 34
μ_f	Prediction of the Gaussian Process, p. 38, 137
μ_k^f	Zeroth- and first-order input-affine modifier of function f , p. 136
∇J_{qa}	Gradient of the cost function obtained from the quadratic approximation, p. 35
$\nabla \mathbf{G}_{qa}$	Gradient of the constraints obtained from the quadratic approximation, p. 35

$\nabla_r \mathcal{J}$	Reduced gradient vector of the economic objective function, p. 16
$\nabla_r \mathcal{J}_{id}$	Reduced gradient vector of the identification objective function, p. 16
$\nabla_r^2 \mathcal{J}$	Reduced Hessian matrix of the economic objective function, p. 16
$\nabla_r^2 \mathcal{J}_{id}$	Reduced Hessian matrix of the identification problem, p. 16
$\nabla_r^2 \mathcal{L}$	Reduced Hessian of the model's Lagrangian function, p. 9
$\nabla_r^2 \mathcal{L}_p$	Reduced Hessian of the plant's Lagrangian function, p. 9
ν	White noise, p. 38
ω	Scaling parameter, p. 43
$\phi(\cdot)$	Standard normal probability function, p. 41
ϕ_i	Individual probability value, p. 43
ρ_k	Performance criteria, p. 32
$\rho_{ad,k}$	Accuracy of the fitted quadratic approximation, p. 36
$\rho_{g,i}$	Gas density of well i , p. 119
ρ_k	Accuracy of the process model, p. 36
$\rho_{mix,i}$	Riser fluid density of well i , p. 119
σ^2	Variance of the exploration noise, p. 151
σ_n^2	Magnitude of the covariance, p. 38
σ_ν^2	Variance of the white noise, p. 38, 137
σ_f^2	Uncertainty around the GP predictor, p. 38, 137
σ_{max}	Upper bound of the process model Hessian spectral radius, p. 31
$\text{Im}(\cdot)$	Imaginary term of the complex number (\cdot) , p. 81, 113
$\tilde{\mathbf{G}}_u$	Combined measurement Jacobian matrix with respect to inputs and parameters, p. 61

$\tilde{\mathbf{Q}}$	Augmented process noise covariance matrix, p. 69
ε	Positive constant, p. 18
$\varphi(\mathcal{U}_d)$	Minimum angle between all possible vectors defined by $(\mathbf{u} - \mathbf{u}^k)$, p. 35
φ	Lower bound to the inverse of the condition number $\kappa_k(\mathbf{u})$, p. 31
b_i	Tuning parameter of the exponential filter associated with constraint i , p. 21
b_k	Constant term defining the hyperplane \mathcal{H}_k , p. 30
c	Constant mean function, p. 38, 137
d	Metric on the metric space \mathbb{U} , p. 17
$f(\cdot)$	Unknown function, p. 38
f_L	Tuning parameter serving as a target for the objective function, p. 41
g_i	Inequality constraints i of the model's economic optimization problem, p. 9, 135
$g_{p,i}$	Inequality constraints i of the plant's economic optimization problem, p. 8
$k(\cdot, \cdot)$	Squared-exponential kernel function, p. 38, 137
$m(\cdot)$	Mean function, p. 38
$m_{g,i}$	Gas holdup of well i , p. 119
$m_{l,i}$	Liquid holdup of well i , p. 119
n_θ	Number of model parameters, p. 9, 68, 111
n_a	Number of actions of the RL-RBF controller, p. 150
n_d	Number of model parameters and unmeasured disturbances, p. 79
n_r	Number of privileged input directions, p. 27
n_s	Number of states of the RL-RBF controller, p. 150

n_u	Number of optimization decision variables, p. 8, 68, 79, 135
n_w	Number of wells, p. 119
n_x	Number of state variables, p. 9, 68, 79, 111
n_y	Number of predicted output variables, p. 8, 9, 68, 79, 111, 135
n_y^a	Number of active constraints, p. 90
n_Λ	Dimension of Λ , p. 24
r	Immediate reward, p. 151
$r(\mathbf{u}, \mathbf{U})$	Correlation vector, p. 38
w_i	Williams-Otto mass fraction of $i \in [A, B, C, E, P, G]$, p. 70, 92, 142
$w_{g,i}^{out}$	Gas production flow rate of well i , p. 119
$w_{l,i}$	Liquid injection flow rate of well i , p. 119
$w_{l,i}^{out}$	Liquid production flow rate of well i , p. 119

List of Abbreviations

APC	Advanced process control, p. 2
ARX	Autoregressive Exogenous, p. 53
CAPE	Computer-aided process engineering, p. 1
CA	Constraint adaptation, p. 4
D-MA	Directional Modifier adaptation, p. 27
DAE	Differential Algebraic Equation, p. 50
DCS	Digital Control System, p. 44
EI	Expected improvement, p. 41
EKF	Extended Kalman filter, p. 3
EMPC	Economic Model Predictive Control, p. 58
ESC	Extremum-Seeking Control, p. 14
FDA	Finite Difference Approach, p. 30
GA	Genetic Algorithm, p. 48
GOPS	Geometry Optimizing Point Selection, p. 34
GP	Gaussian process, p. 37
HRTO	Hybrid real-time Optimization, p. 3
IDF	Implicit Dynamic Feedback, p. 50
ISOPE	Integrated System Optimization and Parameter Estimation, p. 13
KKT	Karush-Kuhn-Tucker, p. 8
LCB	Lower confidence bound, p. 41

MAWQA	Modifier Adaptation with Quadratic Approximation, p. 34
MA	Modifier adaptation, p. 4
MAY	Output Modifier adaptation, p. 26
MHE	Moving Horizon Estimation, p. 50
MISO	Multi-input single-output, p. 52
ML	Machine learning, p. 4
MPC	Model predictive control, p. 2
MU	Multiple-Unit, p. 54
NAPS	Nearest Axis Point Separation, p. 36
NCO	Necessary condition of optimality, p. 8
NEC	Neighboring-Extremal Control, p. 14
NE	Neighboring Extremals, p. 54
NMPC	Nonlinear Model Predictive Control, p. 48
PID	Proportional integral derivative, p. 2
PRBS	Pseudo-random Binary Sequence, p. 52
PSE	Process systems engineering, p. 1
PSO	Particle Swarm Optimization, p. 48
RBFN	Radial Basis Function Network, p. 42
RLS	Recursive Least Square, p. 53
RL	Reinforcement Learning, p. 14
ROPA	RTO with Persistent Parameter Adaptation, p. 3
RTE	Real-time Evolution, p. 48
RTO	Real-time optimization, p. 1
SA	Stochastic Approximation, p. 37
SC	supervisory control, p. 1

SOC	Self-Optimizing Control, p. 14
SSD	Steady-state detection, p. 1
UKF	Unscented Kalman Filer, p. 50

Chapter 1

Introduction

In the field of process system engineering (PSE), real-time optimization (RTO) is a well-known technique of adaptive optimization algorithms that are applied to optimize process operations in real-time. In a very general way, these algorithms have the goal of achieving a certain number of objectives, often economic, such as minimizing cost or maximizing profit, while respecting a set of constraints, which can be related to the process physics, operational limits, safety, and so on. In the real-time approach, the optimization algorithm is executed in an iterative way in order to cope with the process changes, price fluctuations, and disturbances. Although a complete RTO framework is composed of several elements and calculation stages, the mathematical process model is its main component. This model is often a rigorous first-principles model, but it could also be a data-driven one or a combination of first-principles and data-driven models, the so-called hybrid models.

In the context of computer-aided process engineering (CAPE), RTO is not a new technology. The first successful algorithm that found broad acceptance in the industry was proposed by Jang and coworkers in 1987 (JANG *et al.*, 1987). It consists of four main stages:

- i. Steady-state detection (SSD);
- ii. Simultaneous data reconciliation and parameter estimation;
- iii. Model updating;
- iv. Economic optimization with the updated model.

As a static process model is used in the main stages of the algorithm, ii-iv, its use is limited to continuous processes where the operation results in a clear steady-state point, otherwise, the optimal solution of the static economic optimization would never be achievable by the plant.

Another important concern is how to implement the optimal solution in the plant. Sending a new setpoint simultaneously for all optimizing variables to the plant would trigger a dynamic behavior that is not known beforehand since RTO has no dynamic information. If this is done indiscriminately, it could result in dynamic trajectories that could exceed the variables' safety limits, leading to unsafe operation and possible shutdown of the whole plant. For this reason, RTO is often included as one layer in the hierarchical control structure, in which, a supervisory control (SC) layer and a regulatory control layer are present to support the decision-making process (MARLIN and HRYMAK, 1997; NAYSMITH and DOUGLAS, 1995).

In a typical hierarchical control structure, the regulatory control layer is composed of PID loops to reject the high-frequency disturbances while manipulating directly the final control elements, such as valve actuators, and pump drives. Above it, there is the SC layer, also called as advanced process control (APC) layer, which is responsible for making decisions over the setpoints of the PID controllers. Usually, the SC layer uses strategies able to cope with the cross-coupled multi-variable system dynamics in order to reject low-frequency disturbances and sudden changes in multiple setpoints. The most used strategy in the industry is model predictive control (MPC), which is a model-based optimization problem to minimize the quadratic deviation between the predictions of an approximated dynamic model (typically linear) and reference values while respecting a set of constraints. These reference values are typically desired operational points or, in the case where an RTO layer is present, they are the optimal solution of the economic optimization problem (DARBY *et al.*, 2011).

When the process has no SC layer, it is also a common industrial practice to have an RTO in an open loop as an advisor system for guiding the operation team to achieve the process goals (PANTELIDES and RENFRO, 2013). It is important to emphasize that, in any case, the success of any RTO project is extremely dependent on the close participation of the operating team to guarantee the long-term benefit of the application (DARBY *et al.*, 2011).

Over the years, companies have faced more and more pressure to enhance operational efficiency due to an increasingly competitive global market. The general goal is to produce more using less, hence there is a need for reducing feedstock waste, better use of energy, higher process efficiency, and productivity increase. Moreover, with the aggravation of the environmental crisis, society has been pressuring governments for more stringent environmental standards, which are imposed on companies to reduce their environmental impact, especially related to reducing carbon emissions, the by-production of polluting waste, and effluents. The environmental goals and restrictions can also be part of the RTO problem ei-

ther in the objective function or in the constraints. Therefore, the importance of reliable control and optimization layers has increased over the past few decades.

Although most industrial operating teams are well experienced, the operation of any process subjected to so many different objectives is a challenging activity. Often, the optimal operation is close to the bounds of the constraints, which can result in a significant economic loss if the active constraints are not operated within their optimal setpoint. Alternatively, a well-designed control strategy, with an RTO layer, is able to conciliate all the different objectives and guarantee a safe operation (GRACIANO *et al.*, 2015). Despite this fact, there are still several challenges to be tackled in the RTO design that deserve close attention. The following topics are of particular interest to this thesis:

- i. Use of transient measurement;
- ii. Model adequacy;
- iii. Model structural uncertainty.

Recently, different research groups have proposed to incorporate transient measurements in a static RTO framework by replacing the stage of simultaneous data reconciliation and parameter estimation with a dynamic Bayesian observer, such as the extended Kalman filter (EKF) (KRISHNAMOORTHY *et al.*, 2018b; MATIAS and LE ROUX, 2018; VALLURU *et al.*, 2015). This strategy has been called hybrid RTO (HRTO) or RTO with persistent parameter adaptation (ROPA). It is an interesting approach with high potential for real applications. However, there are still some limitations that must be addressed. To name a few, the concern of updating a static model with dynamic parameters data (KRISHNAMOORTHY *et al.*, 2018b), the kind of systems that can benefit from this strategy (VALLURU and PATWARDHAN, 2019), the dynamic behaviors that can be covered and the ones that cannot (CURVELO *et al.*, 2021), the HRTO run frequency (MATIAS and LE ROUX, 2018), and the requirement of having a dynamic process model available to enable its application (DELOU *et al.*, 2021b,c). Here, the former one is considered the bottleneck for an industrial HRTO application, as no typical static RTO design project contemplates the development of a dynamic model, and including it could significantly increase the time and cost of this kind of project.

Another important topic, which has been the focus of research in the RTO field since the 90s, is the model adequacy issue. The main idea is that the static process model must guarantee the observability of the uncertain parameters and unmeasured disturbances to the point that, when the model is updated it will result in an economically optimal solution that matches the plant's optimal solution, under the hypothesis that the model is detailed enough to do so (FORBES and MARLIN, 1994; FORBES *et al.*, 1994). However, this hypothesis is not always true since

the modeling activity naturally involves approximations, and many times it could present unmodeled phenomena, badly posed by modeling assumptions, or disturbances not considered, for example. In these cases, a mismatch between the model and plant arises due to model structural uncertainty (MARCHETTI *et al.*, 2016).

There are several approaches in the literature to overcome model structural uncertainty, such as the integrated system optimization and parameter estimation (ISOPE) (ROBERTS, 1979), the constraint adaptation (CA) (CHACHUAT *et al.*, 2008a; MARCHETTI *et al.*, 2007) and the modifier adaptation (MA) (MARCHETTI *et al.*, 2009; TATJEWSKI, 2002). In general, they all include correction terms to the optimization problem that aim to shift the model-based optimization solution toward the true plant optimal solution. Some of these methods, e.g. CA, MA, and variations, require estimating plant gradients from noisy measurements, which is not an easy task and there is no current reliable method. Due to these reasons, these methods find little space for real industrial applications. However, a few research groups have recently proposed to make use of data-driven models, such as regression models (GAO *et al.*, 2016b) and machine learning (ML) models (DE AVILA FERREIRA *et al.*, 2018; MATIAS and JÄSCHKE, 2019) to fit the model-plant mismatch and to use this information in the economic optimization to compensate the effect of the structural modeling errors.

An alternative to the classic control hierarchy, where the optimization and control layers are separate, is integrating the economic goals into a single optimizing control layer. The so-called economic MPC (DE GOUVÊA and ODLOAK, 1998) is a model-based optimization method that introduces an economic term to the MPC objective function or a tracking term to drive the economic objective function gradient to zero (DE SOUZA *et al.*, 2010). In the RTO implicit/direct methods, an optimization problem is not solved online. In such methods, the economic goals are translated into control objectives and handled by the direct manipulation of the input. Some examples of such methods are the extremum-seeking control (MOROSANOV, 1957), the neighboring-extremal control (PESCH, 1989), the NCO tracking (FRANÇOIS *et al.*, 2005; SRINIVASAN *et al.*, 2003), the self-optimizing control (SKOGESTAD, 2000), and Reinforcement Learning (HOSKINS and HIMMELBLAU, 1992). These methods lack direct handling process constraints, there is usually an extra strategy to deal with changes in the active set of constraints.

1.1 Objective

The main objective of this thesis is to investigate the use of transient data in static RTO frameworks and to propose a way of implementing a Hybrid RTO without

the requirement of having a rigorous dynamic process model. The main assumption is that a rigorous static process model is available, but a rigorous dynamic process model is not. Furthermore, the development of a rigorous dynamic process model is avoided for the sake of minimizing the costs associated with an HRTO design project. The removal of such a requirement is believed to potentially increase the HRTO applicability in large-scale industrial processes.

1.2 Thesis structure

This thesis is organized as a collection of papers that were produced and published during the doctoral candidacy. A common Introduction and Literature Review are presented in Chapters 1 and 2, respectively. Chapters 3 to 7 present a version of the produced papers that were either published in academic journals or conferences. The contributions of each paper are presented in the next section. Finally, Chapters 8 and 9 present overall final remarks and suggestions for future research, respectively.

1.3 Main scientific contributions of the thesis

The main scientific contributions of this work are the new methodologies and approaches in the context of real-time optimization algorithms. The worth-mentioning contributions are topic-wise summarized below. All of them are well in-depth discussed in Chapters 3 to 7.

Chapter 3:

- A new dynamic Hammerstein structure that can be used adequately in the dynamic state and parameter estimation stage of an HRTO architecture, removing the availability requirement of a rigorous first-principle dynamic model;
- Proposition of three HRTO architectures based on the new dynamic Hammerstein structure, one with fixed dynamic matrix, one with a simultaneous adaptation of the dynamic matrix in the Hammerstein-EKF approach, and the last with a separate adaptation of the dynamic matrix in a Recursive Extended Least Squares (RELS) estimator with forgetting factor;
- Demonstration of the performance of the proposed architectures in a CSTR case study, comparing them with the original HRTO using the rigorous dynamic model and the classic RTO with steady-state detection stage.

Chapter 4:

- A new adaptive self-optimizing infinite-horizon MPC (IHMPC) based on the previously proposed dynamic Hammerstein structure;
- A new reconfiguration of the self-optimizing H-matrix algorithm under changes in the active set of constraints is proposed;
- The proposition of a complete HRTTO framework coupled with the proposed adaptive self-optimizing IHMPC, that presents full model compatibility between all control layers by the use of the dynamic Hammerstein structure, no requirement of a first-principle dynamic model, no steady-state wait time, naturally handling changes in the active set of constraints, and finally, the capability of achieving minimum possible economic loss even in the scenario of inaccurate disturbance estimation by the use of self-optimizing control concepts;
- The proposition is demonstrated in a CSTR case study, where two Hammerstein-EKF tunings are compared with the RTO framework proposed by GRACIANO *et al.* (2015).

Chapter 5:

- The application of the Hammerstein RTO/ROPA in a lab-scale experimental rig, which proved the concept of the previously proposed methodology in an actual lab setup. This result opens the possibilities for large-scale systems and increases the industrial applicability of Hybrid RTO strategies.
- The demonstration that the static model can be directly used in the EKF algorithms when the system presents fairly fast dynamics. For this kind of system, there is no need to develop a dynamic mechanistic model, and the Hammerstein RTO/ROPA can be directly applied.

Chapter 6:

- Proposition of a Gaussian Process (GP) based output modifier adaptation (MAy-GP) considering trust-region concepts and comparison;
- A new nonlinear MPC (NMPC) formulation based on an extension of the previously proposed dynamic Hammerstein structure by combining the available static model with the MAy-GP correction terms in the presence of plant-model mismatch;

- Comparison of the proposed MAy-GP with the previously proposed GP-based modifier adaptation (MA-GP) in a CSTR case study in the presence of plant-model mismatch;
- Demonstration of the proposed Hammerstein-NMPC with MAy-GP correction terms in the same CSTR case study in the presence of a plant-model mismatch.

Chapter 7:

- The comparison of a Reinforcement Learning actor-critic-based controller using linear combinations of radial basis functions as an alternative to neural networks with a nonlinear model predictive controller coupled with an Extended Kalman filter in the Van de Vusse case study;
- A hyperbolic function is proposed to work as a maximization activator term once the control goal is satisfied in the reward function, as opposed to the previously proposed logistic activator function.
- A thorough sensitivity analysis on some of the RL training parameters is done to guide practitioners in the training process. The effect of the number of training episodes, the effect of the training exploration, and the effect of the maximization term width and weight in the reward function are presented.

Chapter 2

Literature Review

2.1 Steady-state Optimization Formulation

Given a continuous process in the steady-state condition, observed by a set of output variables, and an existing set of decision variables, the optimization problem consists of determining the optimal decision variables that minimize a certain performance index, or cost function, subject to a set of constraints. The static optimization of a process, or a plant, can be mathematically formulated as follows:

$$\begin{aligned} \mathbf{u}_p^* = \arg \min_{\mathbf{u}} \quad & \mathcal{J}_p(\mathbf{u}) := \mathcal{J}(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \\ \text{s.t.} \quad & G_{p,i}(\mathbf{u}) := g_{p,i}(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.1)$$

in which, $\mathbf{u} \in \mathbb{R}^{n_u}$ are the decision variables of the problem, also called input variables; $\mathbf{y}_p \in \mathbb{R}^{n_y}$ are the measured output variables; $\mathcal{J} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ is the cost function to be minimized; and $g_{p,i} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ for $i \in [1, n_g]$ are the inequality constraints dependent on the input and output variables.

Considering that the gradients of the active constraints set are linearly independent and that functions \mathcal{J}_p and \mathbf{G}_p are continuous and differentiable at \mathbf{u}_p^* , there is a unique set of Lagrangian multipliers $\boldsymbol{\mu}_p^*$ such that the first-order necessary condition of optimality (NCO) or Karush-Kuhn-Tucker (KKT) condition holds (BIEGLER, 2010):

$$\mathcal{L}_p(\mathbf{u}, \boldsymbol{\mu}_p) := \mathcal{J}_p(\mathbf{u}) + \boldsymbol{\mu}_p^T \mathbf{G}_p \quad (2.2a)$$

$$\nabla_{\mathbf{u}} \mathcal{L}_p = \nabla_{\mathbf{u}} \mathcal{J}_p + \nabla_{\mathbf{u}} \mathbf{G}_p \boldsymbol{\mu}_p = 0 \quad (2.2b)$$

$$\mathbf{G}_p \leq 0 \quad (2.2c)$$

$$\boldsymbol{\mu}_p^T \mathbf{G}_p = 0 \quad (2.2d)$$

$$\boldsymbol{\mu}_p \geq 0 \quad (2.2e)$$

In which, \mathcal{L}_p and $\boldsymbol{\mu}_p$ are the Lagrangian function and Lagrangian multipliers of the plant optimization problem, respectively.

Furthermore, the second-order necessary condition of optimality for local minimum states that the reduced Hessian of the Lagrangian function must be positive semi-definite at \mathbf{u}_p^* : $\nabla_r^2 \mathcal{L}_p(\mathbf{u}_p^*) \succeq 0$, while the sufficient condition states that it should be positive definite for a strict local minimum: $\nabla_r^2 \mathcal{L}_p(\mathbf{u}_p^*) \succ 0$. Wherein, the reduced Hessian of the Lagrangian function is defined as:

$$\nabla_r^2 \mathcal{L}_p(\mathbf{u}_p^*) := \mathbf{Z}^T \left[\frac{\partial^2 \mathcal{L}_p}{\partial \mathbf{u}^2}(\mathbf{u}_p^*, \boldsymbol{\mu}_p^*) \right] \mathbf{Z} \quad (2.3)$$

Where \mathbf{Z} is a null-space matrix such that $\left[\frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}(\mathbf{u}_p^*) \right]^T \mathbf{Z} = 0$. The superscript $(\cdot)^a$ indicates the set of active constraints.

This formulation assumes the hypothesis that the cost and constraints functions are known and can be directly calculated from the information of \mathbf{u} and \mathbf{y}_p . However, the real input-output mapping of the plant $\mathbf{y}_p(\mathbf{u})$ is not available, which turns the plant optimization problem formulated in Equation 2.1 to be intractable (MARCHETTI *et al.*, 2016). Frequently, only an approximated steady-state nonlinear model is at hand and not the real input-output mapping of the plant $\mathbf{y}_p(\mathbf{u})$. This model can be mathematically represented as:

$$\mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \quad (2.4a)$$

$$\mathbf{y} = \mathcal{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \quad (2.4b)$$

in which, $\mathbf{x} \in \mathbb{R}^{n_x}$ are the state variables, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ are model parameters and $\mathbf{y} \in \mathbb{R}^{n_y}$ are the predicted output variables. The function $\mathcal{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x}$ is an implicit nonlinear algebraic system of equations that can be solved to obtain the value of \mathbf{x} for a given set of $(\mathbf{u}, \boldsymbol{\theta})$. In addition, the function $\mathcal{H} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_y}$ is an explicit algebraic system of equations. Therefore, for the sake of simplicity, the input-output mapping can be written as:

$$\mathbf{y}(\mathbf{u}, \boldsymbol{\theta}) := \mathcal{H}(\mathbf{x}(\mathbf{u}, \boldsymbol{\theta}), \mathbf{u}, \boldsymbol{\theta}) \quad (2.5)$$

Considering this notation, one can formulate a model-based optimization problem analogous to the plant problem formulated in Equation 2.1:

$$\begin{aligned} \mathbf{u}^* &= \arg \min_{\mathbf{u}} \mathcal{J}(\mathbf{u}, \boldsymbol{\theta}) := \mathcal{J}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \\ \text{s.t. } & G_i(\mathbf{u}, \boldsymbol{\theta}) := g_i(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.6)$$

Analogously to what was shown for the problem in Equation 2.1, the NCO conditions for the model-based optimization problem can be stated as follows:

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\mu}) := \mathcal{J}(\mathbf{u}) + \boldsymbol{\mu}^T \mathbf{G} \quad (2.7a)$$

$$\nabla_{\mathbf{u}} \mathcal{L} = \nabla_{\mathbf{u}} \mathcal{J} + \nabla_{\mathbf{u}} \mathbf{G} \boldsymbol{\mu} = 0 \quad (2.7b)$$

$$\mathbf{G} \leq 0 \quad (2.7c)$$

$$\boldsymbol{\mu}^T \mathbf{G} = 0 \quad (2.7d)$$

$$\boldsymbol{\mu} \geq 0 \quad (2.7e)$$

In which, \mathcal{L} and $\boldsymbol{\mu}$ are the Lagrangian function and multipliers of the model-based optimization problem, respectively.

These conditions also assume that the gradient of the active constraints set is linearly independent, and functions \mathcal{J} and \mathbf{G} are continuous and differentiable at \mathbf{u}^* . In addition, the sufficient condition for a strict local minimum state that $\nabla_r^2 \mathcal{L}(\mathbf{u}^*) \succ 0$.

The problem of plant-model mismatch arises in this kind of model-based formulation because it is desirable that the optimal decision variable of the problem in Equation 2.6, \mathbf{u}^* , to be the same of the problem in Equation 2.1, \mathbf{u}_p^* . If this were true, it would be guaranteed that the solution of the model-based optimization problem would drive the plant to its optimal point, but that does not necessarily happen. In fact, there are many issues related to the model-based optimization problem as it was presented. Besides the optimality problem, which is the capability of the model-based optimum to converge to the plant optimum, the convergence effort is also a decisive matter for real-time applications (QUELHAS *et al.*, 2013) in terms of time duration for convergence. In addition, the feasibility and stability of the RTO architecture are also important features for real-time applications (CHACHUAT *et al.*, 2009; MARCHETTI *et al.*, 2009). Therefore, it is most desired that an RTO architecture presents all the following characteristics to be considered satisfactory for real-time application:

- Optimality;
- Feasibility;
- Stability;
- Fast convergence.

2.2 Static RTO Methods Classification

Several strategies aim to achieve plant optimality. Many authors proposed some sort of classification of the existing static optimization methods (CHACHUAT *et al.*, 2009; ELLIS *et al.*, 1988; FRANÇOIS *et al.*, 2012; MARCHETTI *et al.*, 2016; ZHANG and FORBES, 2008). Here we adopt a modern classification that divides the methods by the role of measurements, use of the models, and uncertainty source (CHACHUAT *et al.*, 2009; MARCHETTI *et al.*, 2016). Figure 2.1 illustrates this classification.

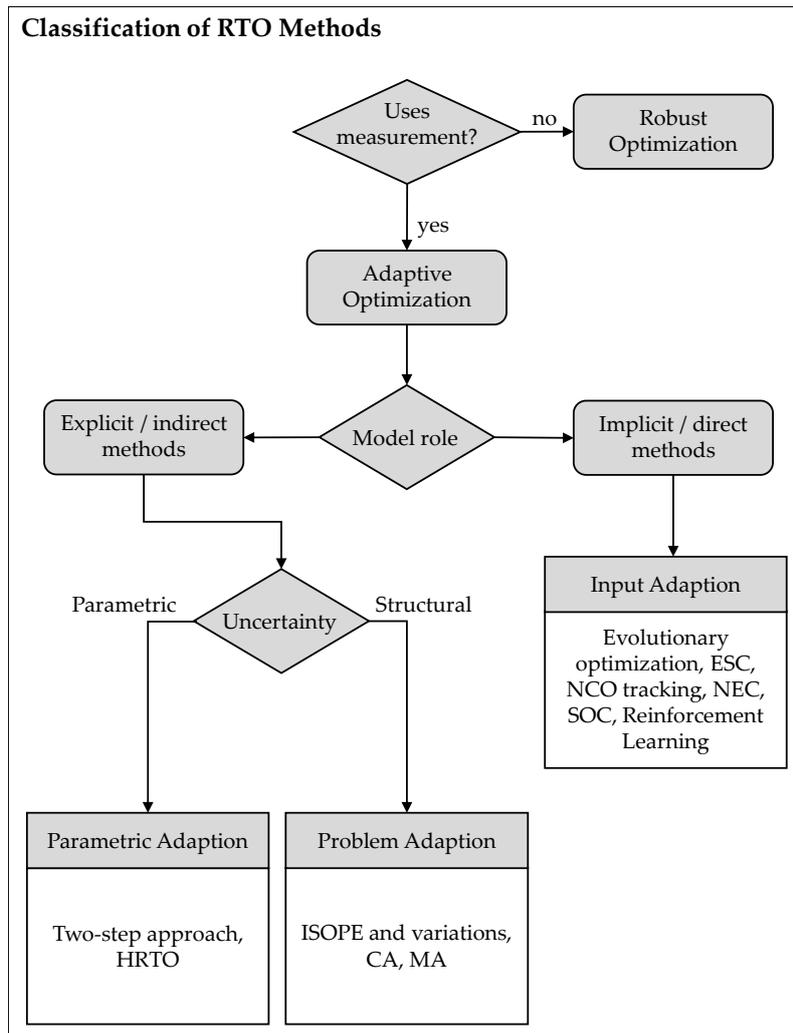


Figure 2.1: RTO classification flowcharts. HRTO: hybrid real-time optimization; ISOPE: integrated system optimization and parameter estimation; CA: constraint adaptation; MA: modifier adaptation; ESC: extremum-seeking control; NCO: Necessary condition of optimality; NEC: neighboring-extremal control; SOC: self-optimizing control.

The major classes of methods are divided by the role of measurements. When measurements are not available, conservatism takes place in a robust optimization

(MÖNNIGMANN and MARQUARDT, 2003). On the other hand, the use of measurements enables the reduction of conservatism by making adjustments to process changes and disturbances (FRANÇOIS *et al.*, 2005). This kind of method is called adaptive optimization. Frequently, the term real-time optimization (RTO) refers solely to this type of method.

The class of the adaptive methods can be further subdivided into the role of the model in the adaptation scheme. When the model is used explicitly in an adaptive optimization problem, the method is called explicit or indirect, otherwise, it is called implicit or direct method. In fact, the use of the model configures the primary source of uncertainty in the majority of RTO techniques (BONVIN, 1998). According to MARCHETTI *et al.* (2016), there are three main sources of uncertainty:

- i. **parametric uncertainty:** values of the model parameters do not correspond to the real process;
- ii. **model structural uncertainty:** oversimplified models, absence of important phenomenon description or neglected dynamics;
- iii. **process disturbances:** unmodeled or unmeasured disturbances.

In order to minimize the detrimental effects of uncertainty, RTO methods make use of measurements in different adaptation strategies. The explicit methods can be divided into two main adaptation approaches. At the level of the model, there is the parametric adaptation, and at the level of the optimization problem there is the problem adaptation:

- i. **parametric adaptation:** in simplified terms, this method needs to have a proper model that satisfies the *model-adequacy condition* (FORBES and MARLIN, 1994; FORBES *et al.*, 1994), and an identification stage is carried out in order to adjust the model parameters so that it could represent the plant satisfactorily in a subsequent optimization stage. This is the procedure of the "two-step approach". According to CHEN and JOSEPH (1987), the algorithm was first proposed by JANG *et al.* (1987) and turned out to be the most widespread RTO algorithm in industrial applications and commercial software (CÂMARA *et al.*, 2016; NAYSMITH and DOUGLAS, 1995; TRIERWEILER, 2014). However, some authors reported the difficulty of the two-step approach to converge to the plant optimum (FRANÇOIS *et al.*, 2005; MATIAS *et al.*, 2018; MENDOZA *et al.*, 2016), to obtain a proper model in an affordable time (BONVIN, 1998), and to achieve and to verify the *model-adequacy conditions* (CHACHUAT *et al.*, 2009; MARCHETTI *et al.*, 2016);

ii. **problem adaptation:** the main idea in this strategy is to add modifiers to the optimization cost function and/or constraints in order to match the KKT conditions of the model-based problem to the plant problem. The first algorithm that appeared in this direction was called integrated system optimization and parameter estimation (ISOPE) (BRDYŚ *et al.*, 1986; ROBERTS, 1979). In fact, ISOPE has also an identification step to estimate model parameters using plant measurements, making its optimization framework similar to the two-step approach. However, plant measurements are also used to estimate the gradients of the plant outputs with respect to the inputs, which enables the computing of the plant cost gradient. These gradients are used in a modified optimization problem in which input-affine terms are added to the problem in Equation 2.6 so that the plant NCO may be satisfied upon the model-based optimization convergence. Another strategy is known as constraint adaptation (CA), in which plant measurements are used to shift the constraints of the original model-based optimization without any parameter estimation step (CHACHUAT *et al.*, 2008a). In fact, an RTO algorithm with a fixed model, i.e. without any model adaptation step, is only possible if measurement-based adaptation is carried out in the cost function and/or in the constraints in a modified optimization framework. Another strategy that makes use of this philosophy is the so-called modifier adaptation (MA), in which input-affine modifiers are added to the optimization cost function and constraints so that plant NCO conditions are satisfied upon convergence (CHACHUAT *et al.*, 2008b; MARCHETTI *et al.*, 2008);

Last, implicit or direct approaches, also known as implicit optimization or optimizing control, use measurement to perform the adaptation directly at the input variables, in a control-inspired way:

iii. **input adaptation:** there are several strategies to perform input adaptation. Evolutionary operation is a sampling technique, where improved steady-state operation periods are successively determined by implementing the Nelder-Mead algorithm to get closer to the optimum (BOX, 1957; BOX and DRAPER, 1969). In extremum-seeking control (ESC), a dynamic perturbation approach is used to estimate plant cost gradient online (ATTA and GUAY, 2019; GUAY and ZHANG, 2003; KRSTIĆ, 2000; KRSTIĆ and WANG, 2000; MARCOS *et al.*, 2004; MOROSANOV, 1957). Neighboring-extremal control (NEC) combines output measurements with model variational analysis to ensure meeting plant NCO (GROS *et al.*, 2006, 2009; PESCH, 1989). In the so-called self-optimizing control (SOC), a sensitivity analysis between model parameters and output measurements is performed to generate the linear com-

binations of the outputs that are invariant to model parameters, so that these combinations can be controlled at their nominal values to reject uncertainty at a minimum acceptable loss (JÄSCHKE *et al.*, 2017; MORARI *et al.*, 1980; SKOGESTAD, 2000). NCO tracking uses measurements to estimate plant NCO and a control algorithm is used to enforce these conditions (FRANÇOIS *et al.*, 2005; JÄSCHKE and SKOGESTAD, 2011; SRINIVASAN and BONVIN, 2007; SRINIVASAN *et al.*, 2008). Finally, a more recent approach, that has been getting attention in the past few years, is the use of self-learning algorithms for control purposes, such as Reinforcement Learning (RL). These algorithms are able to learn with the process operation and adapt their internal empirical models to perform some objectives (BADGWELL *et al.*, 2018; SHIN *et al.*, 2019b; SUTTON and BARTO, 2018a). Although most of the attention is directed to process control, CASSOL *et al.* (2018a) and DELOU *et al.* (2021a) showed that optimization is a promising feature of this kind of algorithm as well.

2.3 The two-step approach

The two-step approach is the most widespread RTO method in industrial applications and commercial software (CÂMARA *et al.*, 2016).

The first step is composed of a static parameter estimation, which is an optimization problem formulated as a simultaneous data reconciliation and parameter estimation problem.

$$\hat{\theta}_k = \arg \min_{\theta} \mathcal{J}_{id} := \|\mathbf{y}_p - \mathbf{y}(\theta)\|_{\mathbf{V}}^2 \quad (2.8)$$

in which, the subscript k denotes the k^{th} run of the RTO, $\hat{\theta}_k$ is the set of optimal parameters, $\mathcal{J}_{id} : \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ is the identification objective function, \mathbf{y}_p is the set of measurements of the output variables, \mathbf{y} is the set of predicted outputs obtained by the steady-state model. \mathbf{V} is a square, diagonal weighting matrix that can be either arbitrarily assigned or based on the covariance of the measurement errors. If \mathbf{V} is arbitrarily assigned, the formulation becomes a weighted least squared estimation; if \mathbf{V} is the measurement covariance matrix, then the formulation becomes a maximum likelihood estimation (BARD, 1970).

The second step is the economic optimization based on the adapted model using the updated parameters:

$$\begin{aligned} \mathbf{u}_k^* = \arg \min_{\mathbf{u}} \quad & \mathcal{J}(\mathbf{u}, \hat{\theta}_k) := J(\mathbf{u}, \mathbf{y}(\mathbf{u}, \hat{\theta}_k)) \\ \text{s.t.} \quad & \mathbf{G}(\mathbf{u}, \hat{\theta}_k) := \mathbf{g}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \hat{\theta}_k)) \leq \mathbf{0} \end{aligned} \quad (2.9)$$

Despite being the most widespread methodology in the industry, the two-step approach requires some caution regarding the assurance of optimality and stability. These subjects are addressed in the following topics. Beyond those, the method presents some vulnerabilities, as discussed by QUELHAS *et al.* (2013), regarding lack of process information, unknown measurement error sources, and numerical aspects of the optimization solvers. Despite those, it has proved economic return in several applications (MARLIN and HRYMAK, 1997). Some successful implementations of the two-step RTO can be found in CAMOLESI *et al.* (2008); CAMPOS *et al.* (2012); DELOU *et al.* (2021d); LIPORACE *et al.* (2009).

2.3.1 Model Adequacy

The problem of model adequacy in a model-based optimization was addressed by BIEGLER *et al.* (1985) by the proposition of some criteria. The authors introduced and compared two optimization problems, one considering a rigorous process model and the other replacing it with a simplified model. They suggested the analysis of the Karush-Kuhn-Tucker (KKT) conditions of both optimal points and argued that the simplified model would be adequate if their optimal points and gradients at the optimal point match with the rigorous model:

$$\begin{aligned} \mathbf{u}^* &= \mathbf{u}_p^* \\ \nabla_{\mathbf{u}} \mathbf{y}(\mathbf{u}^*, \boldsymbol{\theta}) &= \nabla_{\mathbf{u}} \mathbf{y}_p(\mathbf{u}_p^*) \\ \nabla_{\boldsymbol{\theta}} \mathbf{y}(\mathbf{u}^*, \boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \mathbf{y}_p(\mathbf{u}_p^*) \end{aligned} \tag{2.10}$$

However, this criterion is very tight and implies that no simple model would be adequate except the rigorous model, which is not particularly helpful. Also, these conditions are not easy to verify since the optimal points are not previously known to check whether they are KKT points, and, furthermore, estimating the plant gradients from plant noisy data is not a simple task.

Later, FORBES *et al.* (1994) proposed a more appropriate approach for determining model adequacy based on the concept of reduced space (FRANCOIS and BONVIN, 2013), which reduces the optimization problem by removing the equality constraints and the active inequality constraints, remaining only the degrees of freedom that determine the optimum point location within the decision space. This approach also relies on the KKT optimality conditions and is based on the following definition.

Definition 2.3.1 (Point-wise model adequacy). *If \mathbf{u}_p^* is a unique optimum point of the plant, then for a point-wise adequate process model there is at least one set of parameters $\hat{\boldsymbol{\theta}}$ such that the model-based optimization has an optimum at $\mathbf{u}^* = \mathbf{u}_p^*$. In addition, if the*

model-based optimum is unique, then the model is said to be strongly point-wise adequate.

Based on Definition 2.3.1, FORBES *et al.* (1994) developed a simple criterion for point-wise model adequacy which was later extended by FORBES and MARLIN (1994) for a bias-modified problem. However, in essence, model adequacy is proved if there is a set of parameters $\hat{\theta}$ that satisfies the following equations:

$$\nabla_r \mathcal{J}_{id}(\mathbf{u}_p^*, \mathbf{y}(\mathbf{u}_p^*, \hat{\theta})) = \mathbf{0} \quad (2.11a)$$

$$\nabla_r^2 \mathcal{J}_{id}(\mathbf{u}_p^*, \mathbf{y}(\mathbf{u}_p^*, \hat{\theta})) \succ \mathbf{0} \quad (2.11b)$$

$$\nabla_r \mathcal{J}(\mathbf{u}_p^*, \mathbf{y}(\mathbf{u}_p^*, \hat{\theta})) = \mathbf{0} \quad (2.11c)$$

$$\nabla_r^2 \mathcal{J}(\mathbf{u}_p^*, \mathbf{y}(\mathbf{u}_p^*, \hat{\theta})) \succ \mathbf{0} \quad (2.11d)$$

$$G_i(\mathbf{u}_p^*, \mathbf{y}(\mathbf{u}_p^*, \hat{\theta})) \leq \mathbf{0}, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.11e)$$

in which, $\nabla_r \mathcal{J}_{id}$ and $\nabla_r \mathcal{J}$ are the reduced gradient vectors, $\nabla_r^2 \mathcal{J}_{id}$ and $\nabla_r^2 \mathcal{J}$ are the reduced Hessian matrices of the identification problem and the economic objective function, respectively. The criteria imposed by Equations 2.11a and 2.11b are the sufficient conditions for $\hat{\theta}$ being a local strict minimum of the identification problem in the operational point \mathbf{u}_p^* . Equations 2.11c and 2.11d are the sufficient conditions for \mathbf{u}_p^* being a local strict minimum of the economic optimization problem described in Equation 2.6. In addition, Equation 2.11e is the feasibility condition. If all these conditions are satisfied, the plant operational point \mathbf{u}_p^* is a local minimum of the model-based problem for the set of parameters $\hat{\theta}$.

However sufficient, these conditions are not necessary for model adequacy, i.e. the case in which the reduced Hessian matrices are positive semi-definite. Finally, a weakness of these conditions is that they impose the *a priori* knowledge of the plant optimum \mathbf{u}_p^* , which is rarely the case. So, ensuring Equations 2.11a to 2.11e can be an unpractical task (CÂMARA *et al.*, 2016; MARCHETTI *et al.*, 2009).

It is worth mentioning that the definition of a reduced Hessian matrix is given by:

$$\nabla_r^2 \mathcal{J} = \mathbf{Z}^T \left(\frac{\partial^2 \mathcal{J}}{\partial \mathbf{u}^2} \right) \mathbf{Z} \quad (2.12)$$

in which $\mathbf{Z} \in \mathbb{R}^{n_u \times (n_u - n_g^a)}$ is the null space matrix and n_g^a is the number of active inequality constraints.

2.3.2 Stability Conditions

The matter of the two-step RTO stability has great importance for its application in real processes. However, this topic is not broadly exploited in literature. FORBES

and MARLIN (1996) and QUELHAS *et al.* (2013) have considerably contributed to the stability discussion in the two-step approach. Here, some aspects are outlined.

QUELHAS *et al.* (2013) proposed to consider the RTO as a recursive system in terms of \mathbf{u}_k , so the set of optimal solutions can be represented by a nonlinear mapping of the degrees of freedom: $\mathbf{u}_{k+1} = \Gamma(\mathbf{u}_k)$. Then, the stability criteria lie in the fact that the resultant series of the nonlinear mapping must be a Cauchy series, that is, it has guaranteed convergence. Firstly, the concept of contraction must be defined.

Definition 2.3.2 (Contraction). *Given a metric space \mathbb{U} provided with the metric d such that $\mathbb{U} = (\mathbb{U}, d)$, a contraction in \mathbb{U} can be represented by a mapping $\Gamma : \mathbb{U} \rightarrow \mathbb{U}$ if there is a real number $\alpha \in (0; 1)$ such that the following relation is true:*

$$d(\Gamma(\mathbf{u}_n), \Gamma(\mathbf{u}_m)) \leq \alpha d(\mathbf{u}_n, \mathbf{u}_m), \forall \mathbf{u}_n, \mathbf{u}_m \in \mathbb{U} \quad (2.13)$$

in which the lower value of α is known as the Lipschitz constant. With that said, the convergence conditions and the stability conditions can be determined by the Banach fixed-point theorem, stated as follows:

Theorem 2.3.1 (Banach fixed-point theorem). *Given a complete metric space $\mathbb{U} = (\mathbb{U}, d)$ with $\mathbb{U} \subset \mathbb{R}^{n_u}$, if Γ is a contraction in \mathbb{U} , the following are true:*

- i. Γ admits an unique fixed point in \mathbb{U} , giving by \mathbf{u}_∞ such that $\Gamma(\mathbf{u}_\infty) = \mathbf{u}_\infty$;
- ii. for any $\mathbf{u}_0 \in \mathbb{U}$ the series given by $\mathbf{u}_{k+1} = \Gamma(\mathbf{u}_k)$ converges to \mathbf{u}_∞ ;
- iii. the a priori error can be estimated in k by: $d(\mathbf{u}_\infty, \mathbf{u}_k) \leq \frac{\alpha^k}{1 - \alpha} d(\mathbf{u}_1, \mathbf{u}_0)$

With this approach the RTO system can be interpreted as a particular case of the generic fixed point problem of a discrete series, therefore the aforementioned properties can be used to predict the stability conditions and the convergence velocity (QUELHAS *et al.*, 2013).

FORBES and MARLIN (1996) discuss the criteria of point-wise stability in terms of an RTO system, they also consider it to be a recursive series to state the local stability definition:

Definition 2.3.3 (Point-wise stability). *A recursive system of algebraic equations $\mathbf{u}_{k+1} = \Gamma(\mathbf{u}_k)$ is asymptotically stable at a fixed point \mathbf{u}_∞ , if it is Lyapunov stable and there exists a constant $\varepsilon > 0$, such that:*

$$\|\mathbf{u}_\infty - \mathbf{u}_k\| \leq \varepsilon \quad (2.14)$$

therefore,

$$\lim_{k \rightarrow \infty} \|\mathbf{u}_\infty - \mathbf{u}_k\| = 0 \quad (2.15)$$

For Lyapunov stability the following definition takes place:

Definition 2.3.4 (Lyapunov Stability). *Given the metric space $\mathbb{U} = (\mathbb{U}, d)$ and the contraction $\Gamma : \mathbb{U} \rightarrow \mathbb{U}$, a point $\mathbf{u}_\infty \in \mathbb{U}$ is said to be Liapunov stable if there exists a $\beta > 0$ and an $\varepsilon > 0$ so that for any $\mathbf{u}_k \in \mathbb{U}$, $d(\mathbf{u}_\infty, \mathbf{u}_k) < \varepsilon$ and $\forall n \in \mathbb{N}$, $d(\Gamma^n(\mathbf{u}_\infty), \Gamma^n(\mathbf{u}_k)) < \beta$. In which, $\Gamma^n(\mathbf{u}_k)$ represents the n^{th} recursion of Γ , such that $\mathbf{u}_{k+n} = \Gamma^n(\mathbf{u}_k)$.*

With that said, the closed-loop point-wise stability of the two-step RTO approach can be evaluated at the true plant optimum \mathbf{u}_p^* by considering each element (identification, optimization, and plant) an independent nonlinear mapping and taking linear approximations around a small neighborhood of the optimal points (FORBES and MARLIN, 1996):

$$\mathbf{y}_p = \mathbf{y}_p(\mathbf{u}) \quad \therefore \quad \delta \mathbf{y}_p \approx \frac{\partial \mathbf{y}_p^T}{\partial \mathbf{u}}(\mathbf{u}^*) \delta \mathbf{u} \quad (2.16a)$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}(\mathbf{y}_p) \quad \therefore \quad \delta \boldsymbol{\theta} \approx \frac{\partial \boldsymbol{\theta}^T}{\partial \mathbf{y}_p}(\mathbf{y}_p(\mathbf{u}^*)) \delta \mathbf{y}_p \quad (2.16b)$$

$$\mathbf{u}^* = \mathbf{u}^*(\boldsymbol{\theta}) \quad \therefore \quad \delta \mathbf{u}^* \approx \frac{\partial \mathbf{u}^{*T}}{\partial \boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}) \delta \boldsymbol{\theta} \quad (2.16c)$$

Considering that the partial derivatives of the nonlinear mappings exist, the linearized system can be further reduced into the following iterative process:

$$\delta \mathbf{u}_{k+1}^* \approx \frac{\partial \mathbf{u}^{*T}}{\partial \boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}) \frac{\partial \boldsymbol{\theta}^T}{\partial \mathbf{y}_p}(\mathbf{y}_p(\mathbf{u}^*)) \frac{\partial \mathbf{y}_p^T}{\partial \mathbf{u}}(\mathbf{u}^*) \delta \mathbf{u}_k^* \quad (2.17)$$

Therefore, the RTO system is stable at the true plant optimum \mathbf{u}_p^* in the face of small disturbances in the degrees of freedom \mathbf{u} if:

$$\left\| \boldsymbol{\lambda} \left(\frac{\partial \mathbf{u}^{*T}}{\partial \boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}) \frac{\partial \boldsymbol{\theta}^T}{\partial \mathbf{y}_p}(\mathbf{y}_p(\mathbf{u}^*)) \frac{\partial \mathbf{y}_p^T}{\partial \mathbf{u}}(\mathbf{u}^*) \right) \right\| < 1 \quad (2.18)$$

In which, $\boldsymbol{\lambda}(\cdot)$ represents the eigenvalues of the matrix (\cdot) . It is important to highlight that to attend the local stability criterion in Equation 2.18, the sensitivity matrices must be computed in the plant optimum \mathbf{u}^* which is not known *a priori*. Furthermore, the determination of the gradient matrices aforementioned is not trivial and can be subjected to uncertainties, not only associated with measurement but also with the numerical methodology. Finally, this criterion cannot be used to determine global optimum as it is only valid for local optimality.

2.4 The ISOPE approaches

A drawback of the two-step approach previously presented is that it can only ensure plant optimality if the model adequacy conditions are guaranteed, in other words, if the plant-model mismatch source is only parametric. In the presence of structural plant-model mismatch, the methodology may fail to provide an optimal operation. To overcome this issue, several methodologies were then proposed in order to modify the model-based optimization problem to enforce plant optimality. The first noteworthy methodology was the ISOPE, an acronym for Integrated System Optimization and Parameter Estimation, proposed by ROBERTS (1979). This methodology was the first to introduce the concept of modifiers, which would later be the basis for the rise of the MA approaches. The ISOPE is also a two-step approach, but the key element is the inclusion of a gradient correction term to the cost function of the economic optimization problem so that plant optimality is enforced upon convergence.

Several works extended the original proposal of ROBERTS (1979). BRDYŚ *et al.* (1986) extended the methodology to cope with inequality constraints in a constrained optimization problem. ZHANG and ROBERTS (1991) proposed to include the inequality constraint portion directly into the cost function in order to solve an unconstrained optimization problem. Despite being a simpler approach, it generates an infeasible path. TATJEWSKI *et al.* (2001) and TATJEWSKI (2002) proposed to remove the estimation step and ensure the model adequacy by adding a bias correction error between the measurements and the model output variables in an unconstrained case. Expanding this concept, GAO and ENGELL (2005) proposed to use the inequality constraints in an extra optimization step to decide whether an extra input perturbation is required to ensure a better gradient estimation.

2.4.1 Original ISOPE methodology

The original methodology of the ISOPE approach proposed by ROBERTS (1979) was a two-step approach. At a k^{th} RTO iteration, it is considered that the identification step is capable of estimating the set of parameters θ_k so that the following condition is met:

$$\mathbf{y}(\mathbf{u}_k, \theta_k) = \mathbf{y}_p(\mathbf{u}_k) \quad (2.19)$$

This is known as the output-matching condition, which is required in order to match the gradient of the modified cost function to the plant gradient. Then, the

modified economic optimization problem solved is:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_k)) + \boldsymbol{\lambda}_k^T \mathbf{u} \\ \text{s.t. } & \mathbf{u}^{lb} \leq \mathbf{u} \leq \mathbf{u}^{ub} \end{aligned} \quad (2.20)$$

in which, $\boldsymbol{\lambda}_k \in \mathbb{R}^{n_u}$ is the ISOPE modifier and the superscripts lb and ub stands for lower bound and upper bound, respectively. The modifier is updated by the following formula, considering that the gradient of the measurements with respect to the inputs is available:

$$\boldsymbol{\lambda}_k = \left[\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}_k) \right] \frac{\partial J}{\partial \mathbf{y}}(\mathbf{u}_k, \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}_k)) \quad (2.21)$$

Finally, the solution of the problem stated in Equation 2.20 is then filtered in order to determine the new operating point by a first-order exponential filter:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{K}(\mathbf{u}_{k+1}^* - \mathbf{u}_k); \quad (2.22)$$

2.4.2 ISOPE coping with Inequality Constraints

The methodology proposed by BRDYŚ *et al.* (1986) is an expansion of the ISOPE algorithm proposed by ROBERTS (1979) to cope with process-dependent constraints. It is also a two-step approach that assumes the output-matching condition in Equation 2.19. The modified optimization problem is:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_k)) + \boldsymbol{\lambda}_k^T \mathbf{u} \\ \text{s.t. } & g_i(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_k)) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.23)$$

The updated modifier is calculated by:

$$\boldsymbol{\lambda}_k = \left[\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}_k) \right] \left[\frac{\partial J}{\partial \mathbf{y}}(\mathbf{u}_k, \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}_k)) + \frac{\partial \mathbf{g}}{\partial \mathbf{y}}(\mathbf{u}_k, \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}_k)) \boldsymbol{\mu}_k \right] \quad (2.24)$$

in which, $\boldsymbol{\mu}$ is the Lagrangian multipliers of the problem in Equation 2.23 associated with the inequality constraints. The next inputs are calculated by the exponential filter presented in Equation 2.22 and the next Lagrange multipliers are obtained from:

$$\mu_{i,k+1} = \max \{0, \mu_{i,k} + b_i(\mu_{i,k+1}^* - \mu_{i,k})\}, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.25)$$

in which, μ_{k+1}^* is the optimal value of the Lagrangian multipliers and b_i is the tuning parameter of the exponential filter associated with the constraint i .

2.4.3 ISOPE with bias correction term

TATJEWSKI (2002) proposed that the output-mapping condition could be respected without the need to adjust the set of parameters θ by adding a bias correction term in the model outputs, ε_k . With this modification, the RTO algorithm is carried out in a single step with a fixed model.

$$\varepsilon_k := \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}) \quad (2.26)$$

The modified problem with the bias correction term is:

$$\begin{aligned} \mathbf{u}_{k+1}^* = \arg \min_{\mathbf{u}} \quad & J(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}) + \varepsilon_k) + \boldsymbol{\lambda}_k^T \mathbf{u} \\ \text{s.t.} \quad & \mathbf{u}^{lb} \leq \mathbf{u} \leq \mathbf{u}^{ub} \end{aligned} \quad (2.27)$$

The modifiers are calculated by:

$$\boldsymbol{\lambda}_k = \left[\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}) \right] \frac{\partial J}{\partial \mathbf{y}}(\mathbf{u}_k, \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}) + \varepsilon_k) \quad (2.28)$$

Note that this approach can be applied to all ISOPE algorithms that require the output-mapping condition in Equation 2.19. With the proper adjustments, the identification step can be removed by adding the bias correction term in Equation 2.26. With this consideration, the name ISOPE is no longer adequate and the name Modifier adaptation is more appropriate (MARCHETTI *et al.*, 2016). Therefore, the work of TATJEWSKI (2002) was a pioneer in the development of the MA methodology giving birth to a new generation of single-step RTO algorithms. The next section is dedicated to outlining some aspects of the MA methodology.

2.5 Modifier adaptation

The main idea of RTO algorithms based on the MA methodology is to introduce correction terms to modify the model-based economic optimization problem so that it matches plant NCO upon convergence (TATJEWSKI, 2002). The main motivation for MA is that the two-step approach relies on the parameter adjustment of a first-principles model to guarantee plant NCO. However, frequently the model adequacy conditions might not be respected in the face of high disturbances or structural uncertainties, which could lead to sub-optimal and possible infeasible

operation (MARCHETTI *et al.*, 2009). In contrast to the two-step approach, MA does not rely on parameter estimation, instead, it uses measurements to update the modifiers in the successive RTO iterations (MARCHETTI *et al.*, 2016).

In the k^{th} iteration of the RTO, the basic MA approach consists of adding zeroth-order ($\varepsilon_k^J, \varepsilon_k^{g_i} \in \mathbb{R}$) and first-order modifiers ($\lambda_k^J, \lambda_k^{g_i} \in \mathbb{R}^{n_u}$) to the cost function and to the constraints, in contrast with the original proposition of (TAT-JEWSKI, 2002) which considered the correction terms in the output variables. Therefore, the modified optimization problem becomes:

$$\begin{aligned} \mathbf{u}_{k+1}^* = \arg \min_{\mathbf{u}} \quad & \mathcal{J}_m(\mathbf{u}) := J(\mathbf{u}) + (\boldsymbol{\lambda}_k^J)^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^J \\ \text{s.t.} \quad & G_i := g_i(\mathbf{u}) + (\boldsymbol{\lambda}_k^{g_i})^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^{g_i} \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.29)$$

Note that the constant term $\varepsilon_k^J - (\boldsymbol{\lambda}_k^J)^T \mathbf{u}_k$ does not change the optimal solution of the problem in Equation 2.29, therefore it is usually suppressed from the formulation and a more common formulation of the modified cost function considers only the first-order term: $J(\mathbf{u}) + (\boldsymbol{\lambda}_k^J)^T (\mathbf{u})$.

Assuming that the gradients of the cost function and the constraints with respect to the inputs are available, the modifiers can be updated by the following expressions:

$$\varepsilon_k^J = J_p(\mathbf{u}_k) - J(\mathbf{u}_k) \quad (2.30a)$$

$$\varepsilon_k^{g_i} = g_{p,i}(\mathbf{u}_k) - g_i(\mathbf{u}_k), \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.30b)$$

$$\boldsymbol{\lambda}_k^J = \frac{\partial J_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}_k) \quad (2.30c)$$

$$\boldsymbol{\lambda}_k^{g_i} = \frac{\partial g_{p,i}}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial g_i}{\partial \mathbf{u}}(\mathbf{u}_k), \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.30d)$$

The optimal inputs can be directly applied to the plant. However, this strategy can result in aggressive correction and high sensitivity to process noise, which could negatively impact the algorithm's convergence (MARCHETTI *et al.*, 2016). Therefore, it is a common practice to apply first-order exponential filters to update the inputs and the modifiers.

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{K}(\mathbf{u}_{k+1}^* - \mathbf{u}_k) \quad (2.31a)$$

$$\varepsilon_k^g = (\mathbf{I}_{n_g} - \mathbf{K}^\varepsilon) \varepsilon_{k-1}^g + \mathbf{K}^\varepsilon (g_p(\mathbf{u}_k) - g(\mathbf{u}_k)) \quad (2.31b)$$

$$\boldsymbol{\lambda}_k^J = (\mathbf{I}_{n_u} - \mathbf{K}^J) \boldsymbol{\lambda}_{k-1}^J + \mathbf{K}^J \left(\frac{\partial J_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}_k) \right) \quad (2.31c)$$

$$\boldsymbol{\lambda}_k^{g_i} = (\mathbf{I}_{n_u} - \mathbf{K}^{g_i}) \boldsymbol{\lambda}_{k-1}^{g_i} + \mathbf{K}^{g_i} \left(\frac{\partial g_{p,i}}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial g_i}{\partial \mathbf{u}}(\mathbf{u}_k) \right), \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.31d)$$

in which the filter matrices \mathbf{K} , \mathbf{K}^ε , \mathbf{K}^J and \mathbf{K}^{g_i} are usually selected as diagonal matrices with eigenvalues between 0 and 1.

Figure 2.2 illustrates the linear transformations applied in the basic MA scheme.

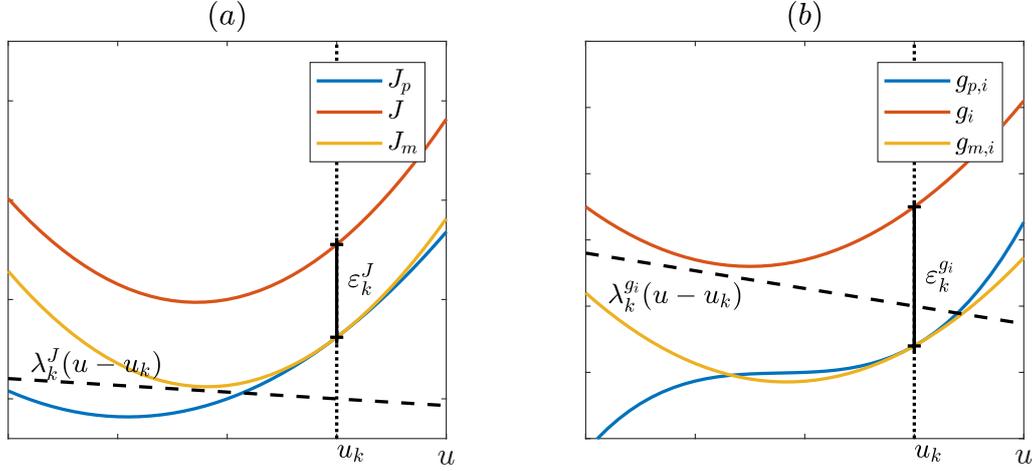


Figure 2.2: Visual representation of the MA approach at k^{th} RTO iteration: (a) Modified cost function; (b) Modified generic constraint i .

Filtering can be applied to any RTO structure, but it is especially found in MA literature as an inheritance from the ISOPE's previous developments. MARCHETTI *et al.* (2016) argue that it can be interpreted as a way of expanding the domain of attraction and a way of dealing with process noise and avoiding large correction steps based on local information.

One can observe that a single run of the MA algorithm would rarely achieve plant optimality, however, the most interesting propriety of the approach is that it has the ability to reach a plant KKT point upon convergence, as stated in the KKT convergence theorem (GAO and ENGELL, 2005):

Theorem 2.5.1 (MA convergence to plant KKT point). *Considering that the input and modifiers are filtered and the gain matrices of the filter are non-singular. If the MA algorithm converges to a fixed point $\mathbf{u}_\infty := \lim_{k \rightarrow \infty} \mathbf{u}_k$ and this point is a KKT point of the modified optimization problem, then \mathbf{u}_∞ is also a KKT point of the plant problem.*

2.5.1 Model Adequacy

As discussed in Section 2.3, FORBES and MARLIN (1996) proposed some criteria for model adequacy in the RTO scheme, which is stated in Definition 2.3.1. MARCHETTI *et al.* (2009) proposed that this definition can be applied to the MA scheme by the observation of the first and second-order plant NCO upon the convergence of the modified model-based optimization problem. MA guarantees first-order KKT conditions upon convergence, as stated in Theorem 2.5.1, then

only the second-order NCO must be assured for model adequacy. That is, ensuring that the reduced Hessian of the Lagrangian is positive semi-definite at \mathbf{u}_p^* . Therefore, MARCHETTI *et al.* (2009) proposed the following conditions:

Definition 2.5.1 (Model adequacy for MA). *Considering that \mathbf{u}_p^* is a regular point for the constraints and a single plant optimum, then:*

- i. *If $\nabla_r^2 \mathcal{L}(\mathbf{u}_p^*) \succ 0$, then the model is adequate;*
- ii. *If $\nabla_r^2 \mathcal{L}(\mathbf{u}_p^*) \succeq 0$, then the second-order condition is not conclusive;*
- iii. *If $\nabla_r^2 \mathcal{L}(\mathbf{u}_p^*) \preceq 0$ or $\nabla_r^2 \mathcal{L}(\mathbf{u}_p^*) \prec 0$, then the model is not adequate.*

2.5.2 Stability Conditions

The stability conditions are frequently discussed in the MA literature as *sufficient conditions for convergence*. Theorem 2.5.1 indicates a property upon convergence, but no consideration is made about the conditions in which the convergence is achieved.

The convergence conditions of the MA scheme lie in the same principles discussed for the two-step approach in Section 2.3, in which the RTO is considered a recursive system that can be represented by a nonlinear mapping: $\mathbf{u}_{k+1} = \Gamma(\mathbf{u}_k)$. Therefore, the convergence conditions are determined by the contraction characteristics of the domain \mathbb{U} .

For the sake of simplifying notation, let $\Lambda \in \mathbb{R}^{n_\Lambda}$ be the collection of all modifiers, where $n_\Lambda = (n_u + 1)(n_g + 1)$:

$$\Lambda_k := (\boldsymbol{\varepsilon}_k^J, (\boldsymbol{\lambda}_k^J)^T, \boldsymbol{\varepsilon}_k^{g_1}, (\boldsymbol{\lambda}_k^{g_1})^T, \dots, \boldsymbol{\varepsilon}_k^{g_{n_g}}, (\boldsymbol{\lambda}_k^{g_{n_g}})^T)^T \quad (2.32)$$

The algorithm represented by Equations 2.29 to 2.31 can be stated as:

$$\mathbf{u}_{k+1} = (1 - \alpha)\mathbf{u}_k + \alpha\mathbf{u}^*(\mathbf{u}_k, \Lambda_k) \quad (2.33)$$

in which, $\mathbf{u}^*(\mathbf{u}_k, \Lambda_k)$ represents the minimizer of the problem in Equation 2.29. With that said, the convergence conditions can be stated as (FAULWASSER and BONVIN, 2014):

Theorem 2.5.2 (Convergence conditions of MA). *Considering the RTO problem in Equation 2.33 and assuming that it is feasible, and has a single minimum, all functions are continuously differentiable on \mathbb{U} and $\alpha \in (0, 1)$. If the mapping $\Gamma(\mathbf{u}) : \mathbf{u} \rightarrow \mathbf{u}^*(\mathbf{u}, \Lambda(\mathbf{u}))$ contracts in the meaning of Definition 2.3.2 and has a fixed point on \mathbb{U} , then the sequence*

\mathbf{u}_k converges to a fixed point:

$$\lim_{k \rightarrow \infty} \|\mathbf{u}^*(\mathbf{u}_k, \Lambda_k) - \mathbf{u}_k\| = 0 \quad (2.34)$$

There are several alternatives to the MA scheme. Then, the following topics are dedicated to outlining these variations.

2.5.3 MA variations

2.5.3.1 Constraint adaptation

The constraint adaptation algorithm was proposed before MA and it is noteworthy that the development of CA made the theoretical foundations for the advent of MA. The adaptation is made by a simple zeroth-order modifier added to the inequality constraints and no modification is carried out in the cost function (CHACHUAT *et al.*, 2008a; MARCHETTI *et al.*, 2007). The CA can be interpreted as a particular case of MA approaches, in which more importance is given to feasibility rather than to optimality. The modified problem of the CA approach is:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}) \\ \text{s.t. } G_i &:= g_i(\mathbf{u}) + \varepsilon_k^{g_i} \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.35)$$

in which each $\varepsilon_k^{g_i}$ is calculated in the same way as the original MA approach and it is also updated with an exponential filter.

It is important to note that CA is not able to converge to a plant KKT point. However, it is able to produce a plant feasible point upon convergence.

This approach was successfully applied to a laboratory scale Solid Oxide Fuel Cell system (MARCHETTI *et al.*, 2011).

2.5.3.2 MA with output modification

This approach recaptures the model shift proposed by TATJEWSKI (2002). The strategy consists of modifying the output variables instead of the cost and constraints by introducing the zeroth-order modifier $\varepsilon_k^y \in \mathbb{R}^{n_y}$ and the first-order modifier $\lambda_k^y \in \mathbb{R}^{n_u \times n_y}$. So the modified problem is defined as:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}, \mathbf{y}_{m,k}(\mathbf{u})) \\ \text{s.t. } g_i &(\mathbf{u}, \mathbf{y}_{m,k}(\mathbf{u})) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\ \mathbf{y}_{m,k}(\mathbf{u}) &= \mathbf{y}(\mathbf{u}) + (\lambda_k^y)^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^y \end{aligned} \quad (2.36)$$

in which, the modifiers are calculated by:

$$\boldsymbol{\varepsilon}_k^y = \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k) \quad (2.37a)$$

$$\boldsymbol{\lambda}_k^y = \frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k) \quad (2.37b)$$

It is important to highlight that this approach also results in a KKT point upon convergence, so Theorem 2.5.1 holds (MARCHETTI *et al.*, 2009).

PAPASAVVAS *et al.* (2019) explored the performance of output MA, called MAy by the authors, in several numeric examples. It is shown that MAy is able to outperform MA when cost and constraint functions are nonlinear with respect to inputs and outputs. Another interesting MAy formulation proposed by PAPASAVVAS and FRANÇOIS (2020) is the addition of a filter-based constraint to the modified optimization problem.

2.5.3.3 MA with Lagrangian modification

This approach is inspired by the ISOPE coping with inequality constraints proposed by BRDYŚ *et al.* (1986). It introduces a first-order modifier to the cost function, $\boldsymbol{\lambda}_k^{\mathcal{L}} \in \mathbb{R}^{n_u}$, and a zeroth-order modifier to the inequality constraint, $\boldsymbol{\varepsilon}_k^{\mathcal{L}} \in \mathbb{R}^{n_g}$. The modified problem is:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}) + (\boldsymbol{\lambda}_k^{\mathcal{L}})^T \mathbf{u} \\ \text{s.t. } &g_i(\mathbf{u}) + \boldsymbol{\varepsilon}_k^{\mathcal{L}} \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.38)$$

in which, the modifiers can be calculated as:

$$\boldsymbol{\varepsilon}_k^{\mathcal{L}} = \mathbf{g}_{p,i}(\mathbf{u}_k) - \mathbf{g}_i(\mathbf{u}_k) \quad (2.39a)$$

$$\boldsymbol{\lambda}_k^{\mathcal{L}} = \frac{\partial \mathcal{L}_p}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\mu}_k) - \frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\mu}_k) \quad (2.39b)$$

Due to the lack of the first-order modifier for the constraints, this approach fails to guarantee convergence to a KKT point. It can result in a slower convergence and more frequent constraint violation before convergence. However, it presents the advantage of requiring only one gradient estimation.

2.5.3.4 Directional MA

The directional modifier adaptation (D-MA) was proposed to reduce the number of plant gradients to be estimated in each RTO iteration (COSTELLO *et al.*, 2015;

COSTELLO *et al.*, 2016). Reliable gradient estimation is still an open challenge in the RTO context (BUNIN *et al.*, 2013), and more aspects of this topic are well reviewed in MARCHETTI *et al.* (2016). Independently of the method, gradient estimation is experimentally expensive and it is the main limiting aspect for MA implementation, as the number of experiments increases linearly with the number of inputs and MA may be impracticable for large-scale processes (MARCHETTI *et al.*, 2016).

The main idea of D-MA is to estimate the gradient only in $n_r < n_u$ privileged input directions and use it for deriving the full gradients. To do so, the matrix $\mathbf{U}_r \in \mathbb{R}^{n_u \times n_r}$ is the matrix of privileged input directions and it is defined as $\mathbf{U}_r = [\delta \mathbf{u}_1 \ \delta \mathbf{u}_2 \ \dots \ \delta \mathbf{u}_r]$. The selection of these directions is typically made of orthonormal vectors that span a linear subspace of n_r dimensions.

The directional derivatives are calculated as:

$$\nabla_{\mathbf{U}_r} J_p := \left. \frac{\partial J_p}{\partial \mathbf{r}}(\mathbf{u}_k + \mathbf{U}_r \mathbf{r}) \right|_{\mathbf{r}=\mathbf{0}} \quad (2.40a)$$

$$\nabla_{\mathbf{U}_r} g_{p,i} := \left. \frac{\partial g_{p,i}}{\partial \mathbf{r}}(\mathbf{u}_k + \mathbf{U}_r \mathbf{r}) \right|_{\mathbf{r}=\mathbf{0}}, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.40b)$$

Hence, the full gradients of the plant can be estimated by:

$$\overline{\nabla} J_k = (\mathbf{I}_{n_u} - \mathbf{U}_r \mathbf{U}_r^\dagger) \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}_k) + (\mathbf{U}_r^\dagger)^T \nabla_{\mathbf{U}_r} J_p \quad (2.41a)$$

$$\overline{\nabla} g_{i,k} = (\mathbf{I}_{n_u} - \mathbf{U}_r \mathbf{U}_r^\dagger) \frac{\partial g_i}{\partial \mathbf{u}}(\mathbf{u}_k) + (\mathbf{U}_r^\dagger)^T \nabla_{\mathbf{U}_r} g_{p,i}, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.41b)$$

in which $\mathbf{r} \in \mathbb{R}^{n_r}$, \mathbf{I}_{n_u} is an n_u -dimensional identity matrix and the $(\cdot)^\dagger$ is the Moore-Penrose pseudo-inverse. That said, the original first-order modifiers presented in Equation 2.30 are altered by replacing the plant gradients with the estimated gradient presented in Equation 2.41.

D-MA is unable to drive the system to a KKT point of the plant upon convergence, therefore Theorem 2.5.1 is not applicable. However, D-MA reaches a point that is optimal for the privileged directions upon convergence (COSTELLO *et al.*, 2016). The great contribution of D-MA is for the case that $n_r \ll n_u$ which would greatly simplify the gradient estimation task.

This approach was successfully applied to an Experimental Crosswind Power Kite system as reported by COSTELLO *et al.* (2016) and COSTELLO *et al.* (2018).

Lastly, the selection of the privileged direction has great importance in a D-MA application and it is still an open matter. SINGHAL *et al.* (2017) proposed an adaptive strategy for each RTO iteration based on parametric perturbation on the Lagrangian gradient to determine more significant directions. SINGHAL *et al.*

(2018) validated this approach in a run-to-run optimization in two different semi-batch reactors. SINGHAL *et al.* (2020) reevaluated the previously proposed approach by suggesting that it is computationally expensive due to the need to compute the matrix of second partial derivatives of the Lagrangian function with respect to model parameters and inputs. So they proposed a new approach based on obtaining gradient values at random parameter samples.

2.5.3.5 Second-order MA

The second-order MA was proposed by FAULWASSER and BONVIN (2014) with the introduction of second-order modifiers into the optimization problem inspired by second-order ISOPE (GOLDEN and YDSTIE, 1989) as a way to assess the sufficient plant NCO. The proposed modified cost and constraints are:

$$\mathcal{J}_m(\mathbf{u}) := J(\mathbf{u}) + (\mathbf{u} - \mathbf{u}_k)^T \Theta_k^J (\mathbf{u} - \mathbf{u}_k) + (\lambda_k^J)^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^J \quad (2.42a)$$

$$G_i := g_i(\mathbf{u}) + (\mathbf{u} - \mathbf{u}_k)^T \Theta_k^{g_i} (\mathbf{u} - \mathbf{u}_k) + (\lambda_k^{g_i})^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^{g_i} \leq 0, \quad (2.42b)$$

$\forall i \in \mathbb{Z}_{\leq n_g}^+$

in which $\Theta_k^J \in \mathbb{R}^{n_u \times n_u}$ and $\Theta_k^{g_i} \in \mathbb{R}^{n_u \times n_u}$ are the second-order modifiers of the cost and constraint, respectively. They can be calculated by the difference between plant and model Hessians:

$$\Theta_k^J := \frac{\partial^2 J_p}{\partial \mathbf{u}^2}(\mathbf{u}_k) - \frac{\partial^2 J}{\partial \mathbf{u}^2}(\mathbf{u}_k) \quad (2.43a)$$

$$\Theta_k^{g_i} := \frac{\partial^2 g_{p,i}}{\partial \mathbf{u}^2}(\mathbf{u}_k) - \frac{\partial^2 g_i}{\partial \mathbf{u}^2}(\mathbf{u}_k), \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.43b)$$

Assuming that the modified second-order MA problem is feasible for all iterations, has a unique optimum and all plant and model functions are twice continuously differentiable at \mathbb{U} , then not only Theorem 2.5.1 holds for a fixed point \mathbf{u}_∞ upon convergence but also:

- i. cost and constraint gradients and Hessians match at \mathbf{u}_∞ ;
- ii. if the problem has a strict local minimum at \mathbf{u}_∞ , then $J_p(\mathbf{u}_\infty)$ is a strict local minimum of $J_p(\mathbf{u})$.

However promising, the second-order MA is even more difficult to implement for real applications due to the difficulties in constructing reliable Hessian approximations from noisy measurements, it is also an open issue in the RTO context.

2.5.3.6 Nested MA

Nested MA was proposed by NAVIA *et al.* (2015) as a way of eliminating the need to estimate plant gradients, instead, a higher-level optimization problem is proposed to directly identify the first-order modifiers. Therefore, the nested MA algorithm is composed of two nested optimization problems: a higher frequency optimization to estimate the inputs at an iteration k ; and a lower frequency derivative-free optimization to estimate the modifiers at an iteration j . The following derivative-free unconstrained optimization problem is solved to calculate first-order modifiers for the modified economic problem:

$$\boldsymbol{\lambda}_{j+1}^* = \arg \min_{\boldsymbol{\lambda}} J_p(\mathbf{u}_\infty^*(\boldsymbol{\lambda}_j)) + (\boldsymbol{\mu}_\infty^*(\boldsymbol{\lambda}_j))^T \mathbf{G}_p(\mathbf{u}_\infty^*) \quad (2.44)$$

in which $\boldsymbol{\lambda}_j$ is the group of first-order modifiers defined by $\boldsymbol{\lambda}_j := [\boldsymbol{\lambda}_j^J \boldsymbol{\lambda}_j^{g_1} \dots \boldsymbol{\lambda}_j^{g_{n_g}}]$, \mathbf{u}_∞^* and $\boldsymbol{\mu}_\infty^*$ are the converged fixed inputs and Lagrangian multipliers, respectively, of the modified economic problem.

The modified economic problem is solved for a fixed j :

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}) + (\boldsymbol{\lambda}_j^J)^T \mathbf{u} \\ \text{s.t. } &g_i(\mathbf{u}) + (\boldsymbol{\lambda}_j^{g_i})^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^{g_i} \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.45)$$

NAVIA *et al.* (2015) showed that the Nested MA has the ability to converge to a plant KKT point, therefore Theorem 2.5.1 holds. However, this convergence is potentially slow due to the many optimization cycles needed.

2.5.3.7 Dual MA

Dual MA arises inspired in the dual ISOPE (BRDYS and TATJEWSKI, 2005) and the main motivation is to ensure reliable gradient estimation. In this formulation, a *duality constraint* is added to the problem to ensure sufficient variability in data for accurate gradient estimation, in which techniques that use measurements of the current and past operating points are employed (MARCHETTI *et al.*, 2016). There are several dual MA propositions, differing mainly in the following aspects:

- i. gradient estimation approach;
- ii. choice of the duality constraint.

Most gradient estimation approaches require computing the matrix of past input differences, more known as the matrix of simplex directions, which is defined by $\mathbf{U}_k := [\mathbf{u}_k - \mathbf{u}_{k-1}, \mathbf{u}_k - \mathbf{u}_{k-2}, \dots, \mathbf{u}_k - \mathbf{u}_{k-n_u}] \in \mathbb{R}^{n_u \times n_u}$. In addition, there is the requirement that \mathbf{U}_k must be a regular matrix.

If the last $n_u - 1$ columns of the matrix of simplex direction at $k + 1$, \mathbf{U}_{k+1} , are linearly independent, then they constitute a basis for the construction of a hyperplane $\mathcal{H}_k = \{\mathbf{u} \in \mathbb{R}^{n_u} : \mathbf{n}_k^T \mathbf{u} = b_k, \text{ where } b_k = \mathbf{n}_k^T \mathbf{u}_k\}$, in which $\mathbf{n}_k \perp \mathcal{H}_k$. So, in order to ensure that \mathbf{U}_{k+1} is regular, the point \mathbf{u}_k must not belong to \mathcal{H}_k . This constraint produces two disjoint feasible regions and, because of that, dual MA schemes solve two modified problems, one for each side of the hyperplane \mathcal{H}_k .

In addition, the following general duality constraint to ensure regularity of \mathbf{U}_{k+1} is added to position the next RTO point taking into consideration the last n_u most recent points:

$$\mathcal{D}_k(\mathbf{u}) := \mathcal{D}(\mathbf{u}, \mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_u+1}) \leq 0 \quad (2.46)$$

Finally, the modified problem of a generic dual MA scheme for each half of hyperplane \mathcal{H}_k are:

$$\begin{aligned} \mathbf{u}_{k+1}^+ = \arg \min_{\mathbf{u}} \quad & \mathcal{J}_m(\mathbf{u}) := J(\mathbf{u}) + (\lambda_k^J)^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^J \\ \text{s.t.} \quad & G_i := g_i(\mathbf{u}) + (\lambda_k^{g_i})^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^{g_i} \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\ & \mathcal{D}_k(\mathbf{u}) \leq 0 \\ & \mathbf{n}_k^T \mathbf{u} \geq b_k \end{aligned} \quad (2.47)$$

and

$$\begin{aligned} \mathbf{u}_{k+1}^- = \arg \min_{\mathbf{u}} \quad & \mathcal{J}_m(\mathbf{u}) := J(\mathbf{u}) + (\lambda_k^J)^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^J \\ \text{s.t.} \quad & G_i := g_i(\mathbf{u}) + (\lambda_k^{g_i})^T (\mathbf{u} - \mathbf{u}_k) + \varepsilon_k^{g_i} \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\ & \mathcal{D}_k(\mathbf{u}) \leq 0 \\ & \mathbf{n}_k^T \mathbf{u} \leq b_k \end{aligned} \quad (2.48)$$

Hence, the next operating point is determined by:

$$\mathbf{u}_{k+1} = \arg \min_{\mathbf{u}} \{ \mathcal{J}_m(\mathbf{u}_{k+1}^+), \mathcal{J}_m(\mathbf{u}_{k+1}^-) \} \quad (2.49)$$

Several authors proposed different approaches to dual MA algorithms:

- i. The original dual ISOPE proposed by BRDYŚ and TATJEWSKI (1994) calculates gradients by finite-difference approach (FDA) and introduces the following duality constraint to avoid ill-conditioning in FDA:

$$\mathcal{D}_k(\mathbf{u}) = \varphi \kappa_k(\mathbf{u}) - 1 \leq 0 \quad (2.50)$$

in which $\kappa_k(\mathbf{u})$ is the condition number at k^{th} iteration of the squared matrix $\bar{\mathbf{U}}_k(\mathbf{u}) := [\mathbf{u} - \mathbf{u}_k, \mathbf{u} - \mathbf{u}_{k-1}, \dots, \mathbf{u} - \mathbf{u}_{k-n_u+1}] \in \mathbb{R}^{n_u \times n_u}$ and φ is a lower bound to the inverse of the condition number $\kappa_k(\mathbf{u})$;

- ii. GAO and ENGELL (2005) also used FDA for gradient estimation and the same approach based on the conditioning number $\kappa_k(\mathbf{u})$. However, they did not use $\mathcal{D}_k(\mathbf{u})$ as a constraint, instead, it is used to determine if additional input perturbation is required. Also, they proposed a methodology to determine the magnitude of this additional perturbation by minimizing the condition number.
- iii. MARCHETTI *et al.* (2010) also used FDA for gradient estimation, but the authors showed that bounding the inverse of the condition number $\kappa_k(\mathbf{u})$ has no effect on the accuracy of the gradient. Therefore, the authors proposed the following dual constraint that bounds the Lagrangian gradient error:

$$\mathcal{D}_k(\mathbf{u}) = \frac{\sigma_{max}}{2} \|\bar{\mathbf{U}}_k^{-1} \text{diag}(\bar{\mathbf{U}}_k \bar{\mathbf{U}}_k^T)\| + \frac{\delta_{noise}}{\ell_{min}} \leq \epsilon_{upper} \quad (2.51)$$

in which the first term copes with truncation error and the second with measurement noise, σ_{max} is an upper bound of the process model Hessian spectral radius, δ_{noise} is the noise level, ℓ_{min} is the nearest complement affine subspaces distance (see MARCHETTI *et al.* (2010) for its complete formulation), and ϵ_{upper} is the desired upper bound of the gradient error norm.

- iv. RODGER and CHACHUAT (2011) calculated the gradients by Broyden's approach and they proposed a dual MA scheme with output modification. The authors observed that a poor estimation in Broyden's approach may lead to a sub-optimal point upon convergence, so they recommended two duality constraints to keep small gradient variance and small step length between iterations, they are, respectively:

$$\mathcal{D}_{1,k}(\mathbf{u}) = (\mathbf{u} - \mathbf{u}_k)^T (\Sigma^T \Sigma)^{-1} (\mathbf{u} - \mathbf{u}_k) \geq 1 \quad (2.52a)$$

$$\mathcal{D}_{2,k}(\mathbf{u}) = (\mathbf{u} - \mathbf{u}_k)^T (\Gamma^T \Gamma)^{-1} (\mathbf{u} - \mathbf{u}_k) \leq 1 \quad (2.52b)$$

in which matrices Σ and Γ should be carefully chosen to avoid making the dual problem infeasible;

- v. NAVIA *et al.* (2012) used the same dual scheme proposed by GAO and ENGELL (2005) to suggest a method to deal with infeasibilities. The authors recommended PI controllers to correct the input variables by controlling the infeasibility error and the error of the inverse of the condition number.

vi. MARCHETTI (2013) introduced a dual MA scheme using linear interpolation to estimate gradients. The authors proposed a different way of calculating the first-order modifiers by accounting for the present and past operating points, approximating the plant in a larger input space. A new duality constraint is added to limit the Lagrangian gradient error estimated in the polyhedral set constituted by the present and past operating points. This constraint presented a larger feasible region, and so, faster convergence:

$$\mathcal{D}_k(\mathbf{u}) = \|\epsilon^f(\mathbf{u}, \mathbf{v}^*)\| + \frac{\delta_{noise}}{\ell_{min}} \leq \epsilon_{upper} \quad (2.53)$$

in which $\epsilon^f(\mathbf{u}, \mathbf{v})$ is the gradient error due to truncation evaluated at point \mathbf{v} , and \mathbf{v}^* is determined as the point that minimizes the gradient error norm within the polyhedral set.

2.5.3.8 MA with Trust-Region formulation

Instead of using filtering to stabilize convergence, BIEGLER *et al.* (2014) and BUNIN (2014) proposed an approach inspired by trust-region methods.

In this method, the following trust-region constraint is added to the modified problem in Equation 2.29:

$$\mathbf{u} \in \mathcal{B}(\mathbf{u}_k, \Delta_k) := \{\mathbf{u} \in \mathbb{R}^{n_u} : |\mathbf{u} - \mathbf{u}_k| \leq \Delta_k\}, \Delta_k \geq 0 \quad (2.54)$$

in which Δ_k is the radius of the ball function $\mathcal{B}(\mathbf{u}_k, \Delta_k)$. This radius is updated based on tuned scalar constants $0 < \eta_1 \leq \eta_2 < 1$ and $0 < \gamma_1 \leq \gamma_2 < 1$ by the following rule:

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2 \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \eta_1 \leq \rho_k < \eta_2 \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k \leq \eta_1 \end{cases} \quad (2.55)$$

The performance criteria, ρ_k , is a measurement of how the plant is performing in relation to model prediction, defined as:

$$\rho_k := \frac{J_p(\mathbf{u}_k) - J_p(\mathbf{u}_{k+1})}{J_{m,k}(\mathbf{u}_k) - J_{m,k}(\mathbf{u}_{k+1})} \quad (2.56)$$

Then, the next input to be applied at the plant is defined as follows:

$$\mathbf{u}_{k+1} := \begin{cases} \mathbf{u}_{k+1}^* & \text{if } \rho_k \geq \eta_1 \\ \mathbf{u}_k & \text{if } \rho_k < \eta_1 \end{cases} \quad (2.57)$$

Note that to compute ρ_k and verify conditions in Equation 2.55 in order to update the ball radius, Δ_k , it is required to apply \mathbf{u}_{k+1} to the plant, and, if condition $\rho_k \leq \eta_1$ is verified, \mathbf{u}_k is then re-applied, as stated in Equation 2.57. Therefore, a too-large trust region can result in successive experiments with no contribution to improving plant cost. This limitation in the nominal case, however, can serve to improve gradient estimation in real applications.

Convergence in this method is observed by the following theorem:

Theorem 2.5.3 (Convergence condition of trust-region based MA (BIEGLER *et al.*, 2014)). *Considering the RTO problem in Equation 2.29 with the addition of the trust-region constraint of Equation 2.54. Assuming that the problem is feasible, and has a single minimum, all functions are continuously differentiable on \mathbb{U} , plant cost function and its Hessian are, respectively, lower- and upper-bounded on \mathbb{R}^{n_u} , and for all k there exists a constant $\kappa \in [0, 1)$ and a sequence $\beta_k > 1$, such that the Cauchy decrease condition is observed:*

$$J_{m,k}(\mathbf{u}_k^*) - J_{m,k}(\mathbf{u}_{k+1}^*) \geq \kappa \|\nabla J_{m,k}(\mathbf{u}_k)\| \min \left\{ \rho_k, \frac{\|\nabla J_{m,k}(\mathbf{u}_k)\|}{\beta_k} \right\} \quad (2.58)$$

then,

$$\lim_{k \rightarrow \infty} \|\nabla J_p(\mathbf{u}_k)\| = 0 \quad (2.59)$$

2.5.4 MA using Approximate Models

2.5.4.1 Convex approximations

FRANÇOIS and BONVIN (2013) proposed the use of convex approximations of the process model in the modified optimization problem in order to ensure model adequacy conditions without prior knowledge of the plant optimum. The authors proposed the construction of the convex approximation for the cost function, J_c , and constraints, G_c , as follows:

$$\begin{aligned} J_c(\mathbf{u}) &= J(\mathbf{u}^*) + \mathbf{a}_J^T(\mathbf{u} - \mathbf{u}^*) + 1/2(\mathbf{u} - \mathbf{u}^*)^T \mathbf{Q}_J(\mathbf{u} - \mathbf{u}^*) \\ G_{i,c}(\mathbf{u}) &= G_i(\mathbf{u}^*) + \mathbf{a}_{G_i}^T(\mathbf{u} - \mathbf{u}^*), \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.60)$$

in which the vector, \mathbf{a}_J and \mathbf{a}_{G_i} and the matrix \mathbf{Q}_J are obtained from the process model by a least squared estimation with constraints to ensure that the matrix \mathbf{Q}_J is symmetric and positive. Unlike other common techniques for constructing convex approximations (FLEURY, 1989; ZHANG and FLEURY, 1997), in which the approximation is updated with the evolution of the process in an SQP-like

approach, here they are kept fixed, and the plant-model mismatch is compensated by the modifiers of the adapted optimization problem.

The authors showed that the scheme is able to drive the system to a KKT point upon convergence, depending on the domain of attraction associated with the initial point. Also, they pointed out that convergence can be delayed depending on the quality of the approximation.

2.5.4.2 Subscript to refer to quadratic approximations

GAO *et al.* (2016b) proposed a framework to estimate plant gradients from noisy measurements by fitting quadratic surrogate models to past operation points. This framework was called Modifier Adaptation with Quadratic Approximation (MAWQA).

The scheme added some interesting features to GAO and ENGELL (2005):

- i. Selection of past points: a proposed screening algorithm to select only a set of points near the actual operating point under some criteria over all collected data;
- ii. Limiting movement: an ellipsoid trust-region constraint is added based on the covariance matrix of the selected points used in the regression, limiting the movement under a space in which the approximation is valid;
- iii. Globalization: switching between model-based and data-driven optimization by tracking process model and quadratic approximated model accuracy.

The proposed screening algorithm, latterly called Geometry Optimizing Point Selection (GOPS), has the objective to ensure that the set of points selected for regression at a k^{th} RTO iteration, $\mathcal{U}_{r,k}$, is well-distributed and has sufficiently distant points. In addition, the authors pointed out that many points selected in the neighborhood of the actual operating point can improve the accuracy of the estimation. Therefore, the regression set is constructed by the union of a neighborhood set and a distant set, $\mathcal{U}_{r,k} = \mathcal{U}_{n,k} \cup \mathcal{U}_{d,k}$, as:

$$\mathcal{U}_{n,k} := \{\mathbf{u} : \|\mathbf{u} - \mathbf{u}_k\| \leq \Delta\mathbf{u}; \mathbf{u} \in \mathbb{U}_c\} \quad (2.61)$$

in which $\Delta\mathbf{u}$ is a screening parameter that controls the size of the neighborhood

ball space, \mathbb{U}_c is the set of all collected past operating points; and,

$$\begin{aligned} \mathcal{U}_{d,k} &:= \arg \min_{\mathcal{U}_d} \frac{1}{\varphi(\mathcal{U}_d)} \sum_{\mathbf{u} \in \mathcal{U}_d} \|\mathbf{u} - \mathbf{u}_k\| \\ \text{s.t. } \text{size}(\mathcal{U}_d) &= \frac{1}{2}(n_u + 1)(n_u + 2) - 1 \\ \mathcal{U}_d &\subset \mathbb{U}_c \setminus \mathcal{U}_{n,k} \end{aligned} \quad (2.62)$$

in which $\varphi(\mathcal{U}_d)$ is the minimum angle between all possible vectors that are defined by $(\mathbf{u} - \mathbf{u}_k)$. The screening optimization problem in Equation 2.62 to obtain the distant set can be approximated by a recursive algorithm presented in GAO *et al.* (2016b). After selecting the n_r past operating point to compose the regression set $\mathcal{U}_{r,k} = \{u_1, u_2, \dots, u_{n_r}\}$, the following $n_g + 1$ regression problems are solved:

$$\min_{\mathcal{P}_f} \sum_{i \in \mathcal{U}_{r,k}} (f_p(\mathbf{u}_i) - f_{qa}(\mathbf{u}_i, \mathcal{P}_f))^2, \quad \forall f \in \{J, g_1, g_2, \dots, g_{n_g}\} \quad (2.63)$$

in which the subscript $(\cdot)_{qa}$ stands for quadratic approximation and $\mathcal{P} = \{a_{1,1}, \dots, a_{n_u, n_u}, b_1, \dots, b_{n_u}, c\}$ is the set of coefficients of the quadratic approximation function as follows:

$$f_{qa}(\mathbf{u}, \mathcal{P}_f) = \sum_{i=1}^{n_u} \sum_{j=1}^i a_{i,j}^f \mathbf{u}_i \mathbf{u}_j + \sum_{i=1}^{n_u} b_i^f \mathbf{u}_i + c^f, \quad \forall f \in \{J, g_1, g_2, \dots, g_{n_g}\} \quad (2.64)$$

With the fitted functions at hand, plant gradients are evaluated at \mathbf{u}_k analytically from the approximations, ∇J_{qa} and ∇G_{qa} . In order to improve the reliability of the estimation, an ellipsoid search space is defined using the covariance matrix of the regression set, $\Sigma_{qa} = \text{cov}(\mathcal{U}_{r,k})$, to define a new constraint to be added to the modified optimization problem in Equation 2.29, such that:

$$\mathbf{u} \in \mathcal{B}_{qa}(\mathbf{u}_k, \Delta_{qa}) := \{\mathbf{u} \in \mathbb{R}^{n_u} : (\mathbf{u} - \mathbf{u}_k)^T \Sigma_{qa}^{-1} (\mathbf{u} - \mathbf{u}_k) \leq \Delta_{qa}\} \quad (2.65)$$

in which Δ_{qa} is a scaling factor of the n_u -axial ellipsoid, $\mathcal{B}_{qa}(\mathbf{u}_k, \Delta_{qa})$, centered at \mathbf{u}_k . The scaling factor can be updated at each iteration in a trust-region-like strategy or it may be kept constant. It is interesting to highlight that the ellipsoid axes are aligned with the eigenvectors of Σ_{qa} and the length of the semi-axes are related to the eigenvalues of Σ_{qa} .

In addition, the accuracies of the model and the regression are tracked in order to define which optimization problem will run at iteration k . The accuracies are

defined as:

$$\rho_k = \max_f \left| 1 - \frac{f_k - f_{k-1}}{f_{p,k} - f_{p,k-1}} \right|, \quad \forall f \in \{J, g_1, g_2, \dots, g_{n_g}\} \quad (2.66a)$$

$$\rho_{ad,k} = \max_f \left| 1 - \frac{f_{ad,k} - f_{ad,k-1}}{f_{p,k} - f_{p,k-1}} \right|, \quad \forall f \in \{J, g_1, g_2, \dots, g_{n_g}\} \quad (2.66b)$$

in which ρ_k is the accuracy of the process model and $\rho_{ad,k}$ is the accuracy of the fitted quadratic approximation. If $\rho_k \leq \rho_{ad,k}$, then the model-based optimization considering the elliptical trust-region constraint is solved. Otherwise, the following data-based optimization is solved:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J_{qa}(\mathbf{u}) \\ \text{s.t. } &g_{qa,i}(\mathbf{u}) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\ &\mathbf{u} \in \mathcal{B}_{qa}(\mathbf{u}_k, \Delta) \end{aligned} \quad (2.67)$$

The authors argued that global convergence is ensured by running a sequence of data-based optimizations, which is a reasonable conclusion based on the assumption that the objective function is convex and bounded from below. However, this is a rather strong assumption. In addition, GAO *et al.* (2016b) argued that, for that reason, the use of convex approximations as proposed by FRANÇOIS and BONVIN (2013) would not be necessary. However, convex approximation was proposed to ensure model adequacy for the model-based problem, and convergence to global optimum is observed if the plant functions are also convex, which is also valid for quadratic approximation which is, in fact, a particular case of convex approximation. Therefore, a more careful investigation should be addressed to this issue in case of non-convexity since the plant convexity assumption is hard to verify and fulfill in practice.

WENZEL *et al.* (2015) proposed another screening algorithm called Nearest Axis Point Separation (NAPS). The algorithm divides the data by the sign configuration and proximity to the $n_u \cdot 2^{n_u}$ segments. NAPS visits each segment selecting the closest points to \mathbf{u}_k until at least $(n_u + 1)(n_u + 2)/2$ points are selected. The performances of NAPS and GOPS were later compared by WENZEL *et al.* (2017). The authors concluded that GOPS outperforms NAPS but with the cost of a higher computational effort. Hence, they propose to use NAPS as a fast pre-screening stage and GOPS as a refinement algorithm. Besides a new screening algorithm, WENZEL *et al.* (2015) also discussed the problem of processes under frequent disturbances. In this case, the authors proposed the inclusion and evaluation of an aging parameter, which, depending on its tuning, would remove old points from the collected data set, \mathbb{U}_c .

GAO *et al.* (2016a) proposed the use of exploitative moves in the MAWQA scheme. This stage has the aim of improving gradient estimation based on the evaluation of the quadratic nature of the plant, measurement noise, and distribution of the regression set. For the case of large dimension problems, HERNÁNDEZ *et al.* (2017) proposed a unified framework that integrates the ideas of directional MA with MAWQA. However, both of the previous propositions require the application of plant perturbations. The first aims to explore the input space, while the second requires the evaluation of process sensitivity in input directions.

HERNÁNDEZ and ENGELL (2017) studied the effect of highly noisy measurements on the MAWQA scheme. They proposed to combine Stochastic Approximation (SA) to exploit the inherent stochastic behavior of the process. Despite MAWQA being able to well handle noisy measurements, the authors showed that their approach presented improved convergence properties for highly noisy measurements.

Finally, HERNANDEZ *et al.* (2018a) and HERNANDEZ *et al.* (2018b) reported a successful application of a real mini plant of a transition metal complex catalyzed hydroformylation process. The authors included steady-state detection and data reconciliation before the MAWQA scheme.

2.5.4.3 Gaussian Process

Gaussian process (GP), or Kriging, regression is a method capable of capturing complex unknown functions even under measurement noise. It is a very common approach used in the field of machine learning, but the first applications in the PSE field were PALMER and REALFF (2002a) and PALMER and REALFF (2002b). These works applied Gaussian process in the context of surrogate optimization, where a GP metamodel is trained on data generated by a physics-based model that would demand too much computational cost for the direct optimization. In the context of RTO, GOMES (2007) was the first to propose the use of GP approximations of the rigorous steady-state model in a real-time iterative optimization approach. The author reported reduced computational cost from the use of the approximation methods when the rigorous model would demand a prohibitive computational cost for a real-time application.

In the context of MA, DE AVILA FERREIRA *et al.* (2018) were the first to propose estimating plant-model mismatch by the use of GP as a higher-order approximation rather than relying on the zeroth- and first-order modifiers.

The idea is to describe an unknown function $f : \mathbb{R}^{n_u} \rightarrow \mathbb{R}$ using some observations that might have a noise part, $y = f(\mathbf{u} + \nu)$, in which $\nu \sim \mathcal{N}(0, \sigma_\nu^2)$ is a white noise with variance σ_ν^2 and zero-mean. GP is a generalized multivariate Gaussian

distribution that can be represented by

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)) \quad (2.68)$$

in which $m(\cdot)$ is a mean function and $k(\cdot, \cdot)$ is a covariance function that takes into account the correlation between the function values. It is common to choose the mean function as a constant, $m(\mathbf{u}) := c$, and the squared-exponential covariance function:

$$k(\mathbf{u}, \bar{\mathbf{u}}) := \sigma_n^2 \exp \left\{ -\frac{1}{2}(\mathbf{u} - \bar{\mathbf{u}})^T \mathbf{\Lambda}(\mathbf{u} - \bar{\mathbf{u}}) \right\} \quad (2.69)$$

in which σ_n^2 is the magnitude of the covariance and the scaling matrix is defined as $\mathbf{\Lambda} := \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_{n_u}\}$.

The regression technique usually employed is the maximum log-likelihood and the predicted distribution of $f(\mathbf{u})$ follows the normal distribution $f(\mathbf{u}) \sim \mathcal{N}(\mu_f(\mathbf{u}), \sigma_f^2(\mathbf{u}))$, with:

$$\begin{aligned} \mu_f(\mathbf{u}) &:= r(\mathbf{u}, \mathbf{U})\mathbf{K}(\mathbf{U})^{-1}\mathbf{y} + c \\ \sigma_f^2(\mathbf{u}) &:= \sigma_n^2 - r(\mathbf{u}, \mathbf{U})\mathbf{K}(\mathbf{U})^{-1}r(\mathbf{u}, \mathbf{U})^T \end{aligned} \quad (2.70)$$

in which the pair (\mathbf{U}, \mathbf{y}) is the input-output data formed by N observations, $K_{i,j}(\mathbf{U}) := k(\mathbf{u}_i, \mathbf{u}_j) + \sigma_v^2 \delta_{i,j}$ for each pair $(i, j) \in 1, \dots, N^2$, $\delta_{i,j}$ is the Kronecker delta function and $r(\mathbf{u}, \mathbf{U}) := [k(\mathbf{u}, \mathbf{u}_1), \dots, k(\mathbf{u}, \mathbf{u}_N)]$. Here, μ_f is interpreted as the prediction of the GP, and σ_f^2 is the uncertainty around this predictor.

DE AVILA FERREIRA *et al.* (2018) proposed $n_g + 1$ MISO GPs to estimate the plant-model mismatch for the cost function and the constraint functions:

$$f_p - f \sim (\mathcal{GP})_f(\mathbf{u}, \mathbf{U}, \mathbf{y}), \quad \forall f \in \{J, g_1, g_2, \dots, g_{n_g}\} \quad (2.71)$$

For the sake of simplifying notation, the indication for the input-output data, (\mathbf{U}, \mathbf{y}) , was dropped in the following discussion. To prevent overfitting, DE AVILA FERREIRA *et al.* (2018) proposed to proceed with the estimation with a limited number of past points within a certain radius from the actual point. However, no more details were discussed on how to obtain a reliable data set, such as the screening algorithm proposed by GAO *et al.* (2016b), or the presence of frequent disturbances.

The authors proposed to solve the following high-order modified problem,

referred here as MA-GP (DE AVILA FERREIRA *et al.*, 2018):

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}) + \mu_J(\mathbf{u}) \\ \text{s.t. } & g_i(\mathbf{u}) + \mu_{g_i}(\mathbf{u}) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.72)$$

The scheme was compared to a standard MA approach showing the great potential of the MA-GP to drive the plant to its optimum point. The MA-GP outperformed the standard MA in convergence time, stability around the optimum, feasible path taken, and suppression of noise. However, very little attention was given to the standard MA implementation and no mathematical foundation is yet available to support this better performance.

DEL RIO CHANONA *et al.* (2019) extended the MA-GP scheme proposed by DE AVILA FERREIRA *et al.* (2018) based on trust-region ideas. They proposed a simple approach to update the radius based on the accuracy of the GP predictors, called MA-GP-ITR:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}) + \mu_J(\mathbf{u}) \\ \text{s.t. } & g_i(\mathbf{u}) + \mu_{g_i}(\mathbf{u}) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\ & \|\mathbf{u} - \mathbf{u}_k\| \leq \Delta_k \end{aligned} \quad (2.73)$$

in which Δ_k is the radius of a ball trust-region constraint, updated considering the constants $0 < \eta_1 \leq \eta_2 < \eta_3 < 1$ and $0 < \gamma_1 < 1 < \gamma_2$, as:

$$\Delta_{k+1} \in \begin{cases} \gamma_1 \Delta_k & \text{if } \rho_{k+1} < \eta_2 \text{ or } g_{p,i}(\mathbf{u}_{k+1}) > 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\ \min\{\gamma_2 \Delta_k, \bar{\Delta}\} & \text{if } \rho_{k+1} > \eta_3 \text{ and } \|\mathbf{u} - \mathbf{u}_k\| = \Delta_k \\ \Delta_k & \text{otherwise} \end{cases} \quad (2.74)$$

in which, $\bar{\Delta}$ is an upper bound for Δ_k and the performance criteria is evaluated as presented in Equation 2.56. It is interesting to highlight that this approach is slightly different from the formulation presented in Section 2.5.3.8, mainly because of the addition of a "backtrail" mechanism in the radius updating strategy to reduce the trust-region size whenever a constraint is violated. This approach is capable of minimizing infeasible paths by taking actions to drive the path back to feasibility whenever a violation is detected. In addition, radius increase is only possible if the trust-region constraint is active.

DEL RIO CHANONA *et al.* (2019) proposed another approach where multiple trust regions are defined based on the GP predictors variance. The approach,

called MA-GP-MPTR, solves the following modified problem:

$$\begin{aligned}
\mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} J(\mathbf{u}) + \mu_J(\mathbf{u}) \\
\text{s.t. } & g_i(\mathbf{u}) + \mu_{g_i}(\mathbf{u}) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\
& \sigma_{J,k}(\mathbf{u}) \leq \Delta_{J,k} \\
& \sigma_{g_i,k}(\mathbf{u}) \leq \Delta_{g_i,k}, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+
\end{aligned} \tag{2.75}$$

in which $\Delta_{f,k}$ are the radius of each separated trust region defined from the standard deviation of each GP predictor considering $f \in \{J, g_1, \dots, g_{n_g}\}$. The radius updating associated with the cost function is:

$$\Delta_{J,k+1} \in \begin{cases} \gamma_1 \Delta_{J,k} & \text{if } \rho_{k+1} < \eta_2 \\ \min\{\gamma_2 \Delta_{J,k}, \bar{\Delta}_J\} & \text{if } \rho_{k+1} > \eta_3 \text{ and } \sigma_{J,k} = \Delta_{J,k} \\ \Delta_k & \text{otherwise} \end{cases} \tag{2.76}$$

And, radius updating associated with each constraint $i \in \mathbb{Z}_{\leq n_g}^+$ is:

$$\Delta_{g_i,k+1} \in \begin{cases} \gamma_1 \Delta_{g_i,k} & \text{if } g_{p,i}(\mathbf{u}_{k+1}) > 0 \\ \min\{\gamma_2 \Delta_{g_i,k}, \bar{\Delta}_{g_i}\} & \text{if } g_{p,i}(\mathbf{u}_{k+1}) < 0 \text{ and } \sigma_{g_i,k} = \Delta_{g_i,k} \\ \Delta_k & \text{otherwise} \end{cases} \tag{2.77}$$

In both proposed algorithms, the input updating criterion is:

$$\mathbf{u}_{k+1} \in \begin{cases} \mathbf{u}_k & \text{if } \rho_{k+1} < \eta_1 \text{ or } g_{p,i}(\mathbf{u}_{k+1}) > 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\ \mathbf{u}_{k+1} & \text{otherwise} \end{cases} \tag{2.78}$$

Note that the evaluation of ρ_{k+1} imposes the need to apply \mathbf{u}_{k+1} to the plant and if the updating criteria does not hold, \mathbf{u}_k is then reapplied. This is an unwanted issue of the algorithm that might produce unnecessary points that do not contribute to cost decreases or are infeasible. In contrast with the trust-region formulation presented in Section 2.5.3.8, the "backtrail" approach would be able to drive the system back to a cost-decreasing and feasible path, no strategy is introduced to prevent spurious solutions from being applied to the plant.

The authors argued the formulation of MA-GP-MPTR is more flexible and resilient to poor scaling in large problems. Also, they highlighted that the dispersed measurements across the entire feasible region can prevent convergence to a local minimum. However, it comes with the cost of adding extra nonconvexity to the problem. The results showed that MA-GP followed an infeasible path, in contrast with the results presented by DE AVILA FERREIRA *et al.* (2018), MA-GP-ITR and MA-GP-MPTR initially violate the constraint but then the "backtrail" strategy was

able to drive them back to a feasible path. All the approaches were able to reach the truly constrained optimum, which confirmed the potential of the GP approach to describe plant-model mismatch even under the presence of measurement noise. The authors claimed that MA-GP-ITR performs faster and is easier to tune than MA-GP, due to the absence of filtering. Also, an advantage of the MA-GP-MPTR is that it guarantees that the extrapolation errors of cost and constraint functions are always under control.

DEL RIO CHANONA *et al.* (2021) applied the concept of acquisition functions to enhance the exploration capabilities of MA-GP. The idea behind the acquisition functions is to select the next acquired point by making use of the past selected points and the uncertainty in the decision space, it provides a tradeoff between exploration and exploitation (SHAHRIARI *et al.*, 2016). The authors proposed the use of the lower confidence bound (LCB) and the expected improvement (EI) functions:

$$\mathcal{A}_{LCB}[\mu_f, \sigma_f](\mathbf{u}) := \mu_f(\mathbf{u}) - \gamma\sigma_f(\mathbf{u}) \quad (2.79)$$

$$\begin{aligned} \mathcal{A}_{EI}[\mu_f, \sigma_f, f_L](\mathbf{u}) &:= -[f_L - \mu_f(\mathbf{u})] \Phi(\mathcal{N}_{EI}) - \sigma_f(\mathbf{u})\phi(\mathcal{N}_{EI}), \\ \mathcal{N}_{EI} &:= \frac{f_L - \mu_f(\mathbf{u})}{\sigma_f(\mathbf{u})} \end{aligned} \quad (2.80)$$

in which γ is an exploration weight, $\Phi(\cdot)$ is the cumulative normal distribution function, $\phi(\cdot)$ is the standard normal probability function and f_L is a tuning parameter that works as a target for the objective function, for which the authors proposed to use the best-observed value.

Thus, the authors proposed to solve the following optimization problem considering the use of an acquisition function:

$$\begin{aligned} \mathbf{u}_{k+1}^* &= \arg \min_{\mathbf{u}} \mathcal{A}[J(\mathbf{u}) + \mu_j^k, \sigma_j^k, \cdot](\mathbf{u}) \\ \text{s.t. } &g_i(\mathbf{u}) + \mu_{g_i}^k(\mathbf{u}) \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \\ &\|\mathbf{u} - \mathbf{u}_k\| \leq \Delta_k \end{aligned} \quad (2.81)$$

in which $\mathcal{A}[\cdot, \cdot, \cdot](\mathbf{u})$ is one of the discussed acquisition functions, lower confidence bound (\mathcal{A}_{LCB}) or expected improvement (\mathcal{A}_{EI}).

DEL RIO CHANONA *et al.* (2021) applied the proposed approach in an illustrative example, in the Williams-Otto reactor, and a batch-to-batch bioreactor. In all of the cases, both acquisition functions presented a very similar performance. In the illustrative example, the authors presented the benefit of using an acquisition function, which shows that the problem converges to the plant's optimum point more frequently than without it, and the benefit of knowing the process

noise, which considerably decreases the GP uncertainty and improves the modified optimization results. Also in the illustrative example, the authors showed the benefit of using the uncertain model as opposed to no model at all, which proved to have reduced the uncertainty of the GP approximation and faster convergence to the plant optimum. In the Williams-Otto reactor, the authors only compared the performance of the use of EI and LCB acquisition functions, but the result considering no additional exploration was not presented. However, no other acquisition functions were tested such as the probability of improvement, Thompson sampling, entropy search, and knowledge-gradient policy (SHAHRIARI *et al.*, 2016). In addition, it has been pointed out that no single function will be capable of addressing every application and, therefore, one might consider using a pool of various acquisition functions (HOFFMAN *et al.*, 2010). However, such an idea was not explored in the context of RTO to this data. Finally, in a batch-to-batch bioreactor, DEL RIO CHANONA *et al.* (2021) compared the approach with and without the knowledge of the process model, which confirmed that the prior use of an uncertain model produced a better GP approximation, faster, and more consistent convergence to the plant's optimal point, although the model-free results demonstrate a potential of a completely data-driven approach using GP approximations.

2.5.4.4 RBF Neural Network

The idea of modeling plant-model mismatch by a Radial Basis Function Network goes back to the work of JOHANSEN and FOSS (1992). However, the introduction of this idea into a MA scheme was only recently proposed by MATIAS and JÄSCHKE (2019). The approach takes advantage of the RBFN structure to extract its gradients analytically.

The main idea is to model the plant cost and constraint functions in two portions, one from the rigorous model and the other the plant-model mismatch, ε , that might be subject to noise, ν :

$$f_p(\mathbf{u}) = f(\mathbf{u}) + \varepsilon_f(\mathbf{u}, \nu), \quad \forall f \in \{J, g_1, g_2, \dots, g_{n_g}\} \quad (2.82)$$

The portion of the plant-model mismatch is then modeled by the RBFN and the plant gradient is estimated by taking the gradient of the model and the gradient of the RBFN approximation:

$$\nabla_{\mathbf{u}} f_p(\mathbf{u}) = \nabla_{\mathbf{u}} f(\mathbf{u}) + \nabla_{\mathbf{u}} \varepsilon_f(\mathbf{u}, \nu), \quad \forall f \in \{J, g_1, g_2, \dots, g_{n_g}\} \quad (2.83)$$

Each of the n_n neurons of an RBFN is a Gaussian-like function centered at a

given input value, $\boldsymbol{\mu}$:

$$\phi_i(\mathbf{u}) = \exp\{-\omega\|\mathbf{u} - \boldsymbol{\mu}_i\|\}, \quad \forall i \in \mathbb{Z}_{\leq n_n}^+ \quad (2.84)$$

in which ω is a scaling parameter that controls the width of the Gaussian curve. To improve accuracy, the activation function is the normalization of all ϕ_i between 0 and 1:

$$r_i(\mathbf{u}) = \frac{\phi_i(\mathbf{u})}{\sum_{j=1}^{n_n} \phi_j(\mathbf{u})}, \quad \forall i \in \mathbb{Z}_{\leq n_n}^+ \quad (2.85)$$

The authors proposed to construct $n_g + 1$ MISO RBFNs by taking the weighted sum of all n_n activation functions applied to each of the n_r points in the regression set for all $f \in \{J, g_1, g_2, \dots, g_{n_g}\}$:

$$\begin{bmatrix} \varepsilon_{f,1} \\ \varepsilon_{f,2} \\ \vdots \\ \varepsilon_{f,n_r} \end{bmatrix} = \begin{bmatrix} 1 & r_{f,1}(\mathbf{u}_1) & r_{f,2}(\mathbf{u}_1) & \dots & r_{f,n_n}(\mathbf{u}_1) \\ 1 & r_{f,1}(\mathbf{u}_2) & r_{f,2}(\mathbf{u}_2) & \dots & r_{f,n_n}(\mathbf{u}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & r_{f,1}(\mathbf{u}_{n_r}) & r_{f,2}(\mathbf{u}_{n_r}) & \dots & r_{f,n_n}(\mathbf{u}_{n_r}) \end{bmatrix} \begin{bmatrix} w_{f,0} \\ w_{f,1} \\ \vdots \\ w_{f,n_n} \end{bmatrix} \quad (2.86)$$

Addressing these matrices as $\mathbf{E}_f = \mathbf{R}_f \cdot \mathbf{W}_f$, it is possible to determine the weights of each RBFN by a recursive least squared estimation as follows:

$$\mathbf{W}_f = (\mathbf{R}_f^T \mathbf{R}_f)^{-1} \mathbf{R}_f^T \mathbf{Y}_f \quad (2.87)$$

The authors show that the gradients of the RBFN can be analytically evaluated by:

$$\nabla_{\mathbf{u}} \varepsilon_f = -2\beta (\bar{\mathbf{W}} \cdot \text{diag}\{\bar{\mathbf{R}}\} \cdot \mathbf{M} - \bar{\mathbf{R}} \cdot \mathbf{M}^T \cdot \bar{\mathbf{W}} \cdot \bar{\mathbf{K}}^T) \quad (2.88)$$

in which

$$\begin{aligned} \mathbf{M}(\mathbf{u}) &= [(\mathbf{u} - \boldsymbol{\mu}_1), (\mathbf{u} - \boldsymbol{\mu}_2), \dots, (\mathbf{u} - \boldsymbol{\mu}_{n_n})]^T \\ \bar{\mathbf{R}}(\mathbf{u}) &= [r_1(\mathbf{u}), r_2(\mathbf{u}), \dots, r_{n_n}(\mathbf{u})]^T \\ \bar{\mathbf{W}}(\mathbf{u}) &= [w_1, w_2, \dots, w_{n_n}]^T \end{aligned} \quad (2.89)$$

On every iteration, the neurons are repositioned in the input space by the K-means clustering strategy SCHWENKER *et al.* (2001) and the network is retrained. Also, a probing policy similar to GAO *et al.* (2016b) is applied. The idea is to guarantee a well-distributed with sufficiently distant points in \mathbf{R}_f and to explore the neighborhood region for increased accuracy of gradient estimation. The authors tested the methodology in a gas-lift well simulation and reported good results

even for a significant amount of noise. However, they did not compare the results to other MA approaches, the standard MA, or the MAWQA (in which this framework is very similar). In addition, the plant-model mismatch introduced to the simulation was only from a parametric source, no structural uncertainty was explored. The approach could be better explored in order to clarify its performance compared to other MA strategies and for a variety of mismatch sources.

2.6 Use of Transient Measurement

As current RTO industrial practice mainly relies on the use of static models in the optimization, data reconciliation, and parameter estimation steps, the requirement of acquiring steady data is mandatory. Otherwise, using transient data directly could inject significant error into the optimization solution, which could lead to a sub-optimal operation or even potential instabilities (ENGELL, 2007). In an attempt to reject transient measurements, an SSD step is included in the classic RTO framework to ensure the subsequent stages are only triggered when a steady state is detected, otherwise, the execution stays on hold. Figure 2.3 illustrates the classic RTO execution steps and how it is located in the control hierarchy.

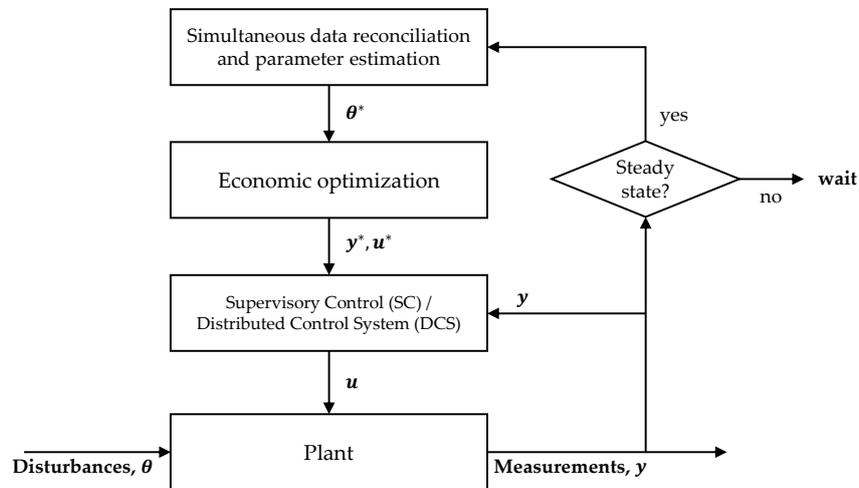


Figure 2.3: Classic static RTO framework scheme.

There are several SSD methodologies available (KELLY and HEDENGREN, 2013; RHINEHART, 2013; TURAN and JÄSCHKE, 2023), and most of them are based on statistical quantities taken from real-time measurements to detect whether the current point represents a steady state with a certain degree of confidence. However, the reality of industrial-scale application shows that moments of steady-state conditions are an idealization rather than a frequent reality. Consequently, the waiting times for RTO executions can be very long, which usually

results in the necessity of tweaking the SSD tuning to allow for more frequent RTO execution, which will ultimately allow for runs to occur during transient moments and potentially degrade the RTO performance.

This becomes even more of an issue in continuous processes with frequent and/or slowly drifting disturbances, which makes SSD even rarer. Moreover, in between every RTO execution, the plant needs to continuously move from one operating point to another which may result in sub-optimal solutions for most of the operation time. In fact, the RTO's long wait time has been reported as one of its major issues for the last three decades (BROOKS, 2003; DARBY *et al.*, 2011; ENGELL, 2007; FRIEDMAN, 2005, 1995, 1998).

2.6.1 Parameter adaptation schemes

In the context of parameter adaptation schemes, such as the two-step approach (JANG *et al.*, 1987), one of the first alternatives to incorporate transient measurements was the proposition of the so-called DRTO where a dynamic model is used in the RTO problem (BIEGLER, 1984a; POLLARD and SARGENT, 1970; SARGENT and SULLIVAN, 1978). The main incentive for using DRTO is that dynamic models allow a more frequent re-optimization of plant economics since there is no need to wait for the plant to reach a steady state. Over the last decades, the DRTO scheme has been applied to a wide variety of chemical processes, for example, processes that exhibit long transient dynamics such as integrated plants with recycle streams (TOSUKHOWONG *et al.*, 2004), bio-ethanol processes (OCHOA *et al.*, 2010), FCC converter unit (ALMEIDA NT and SECCHI, 2011), polymerization industries (PONTES *et al.*, 2015), and autocatalytic esterification reactors (ROHMAN *et al.*, 2019), to name a few. Although extensively adopted in chemical engineering research papers, DRTO has not yet found broad acceptance in the process industry (CÂMARA *et al.*, 2016; CAMPOS *et al.*, 2009), and only a few applications to real systems have been reported in the literature (e.g., AHO *et al.* (2009), HUANG (2010), and MÜLLER *et al.* (2017)). Some of the reasons hindering DRTO implementation in an industrial environment are the fact that the dynamic models are not so common in industry (PISTIKOPOULOS *et al.*, 2021) and that the resulting dynamic nonlinear optimization problem is much more complex than its static counterpart, which may raise several numerical issues in online applications.

Over the years, several authors made efforts to make changes and adjustments to the static two-step approach so that it can incorporate the use of transient data. Moreover, these kinds of frameworks present an economic performance close to the DRTO with much less computational effort for continuous processes (KRISH-

NAMOORTHY *et al.*, 2018b), which has driven significant attention and interest to them. The next paragraphs discuss the main recent developments in the use of transient data in the context of the two-step RTO schemes.

2.6.1.1 Seminal works

One of the first ideas to overcome the steady-state wait time in the context of the two-step RTO was reported by BESL *et al.* (1998) in an application commissioned in 1996 to a light-naphtha isomerization plant. The authors reported that the process presented relatively slow dynamics with a wait time of 3 hours for a steady state after a step disturbance in the distillation reflux flow rate and 8-10 hours after changes in the feed characteristics, which occurred at least twice a day. Therefore, the classic RTO would present a very low frequency and would fail to keep the plant in a near-optimal performance. BESL *et al.* (1998) point out that static data reconciliation and parameter estimation is the critical step where the dynamic data would have the most significant impact. Therefore, the authors proposed to change the classic two-step approach by the inclusion of a second verification loop, in which if the data is not steady only the optimization would run using the offsets and parameters estimated by the last data reconciliation run and the data reconciliation step would only run when the data was found to be in steady state. Their optimization approach could be comparable with an ISOPE in which only the zeroth-order corrections were applied. Figure 2.4 illustrates the RTO framework proposed by BESL *et al.* (1998). With this simple change in the frequency of each step of the classic RTO scheme, the authors reported a productivity increase of 14 % and an economic benefit greater than \$1.5 million per year compared to the classic scheme. However, this methodology would only be valid when all or most of the critical disturbances and parameters are measured or known. Since there are long waiting periods for the model adaptation, the plant parameters would continuously drift between each parameter estimation execution, and the optimization solutions would be sub-optimal for most time of the operation.

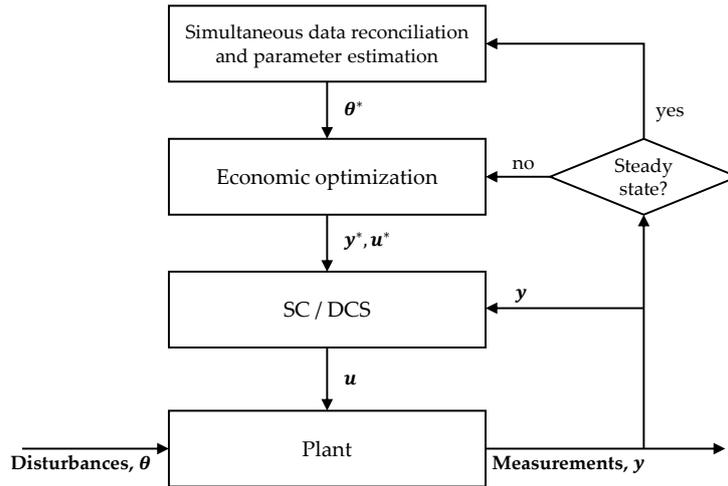


Figure 2.4: Besl's RTO framework scheme (BESL *et al.*, 1998).

PRIOR and LOPEZ (1999) reported an RTO scheme without the use of steady-state detection applied to an ethylene plant. The data reconciliation, parameter estimation, and optimization ran in a modular way, in which none of these steps depended on the others to run. This is made possible by a common database where the steps share information. The parameter estimation was done in three modules, where the model was partitioned to allow for an asynchronous model update, each running within 15-minute period. Figure 2.5 illustrates the RTO framework proposed by PRIOR and LOPEZ (1999).

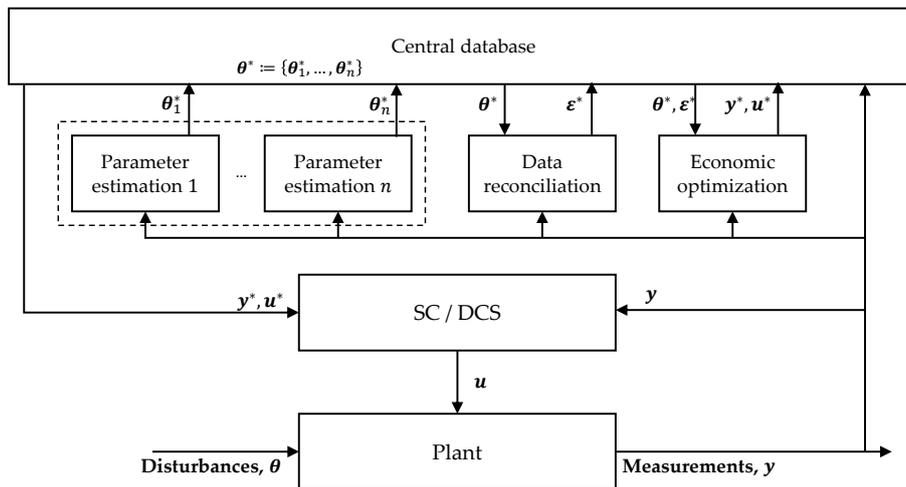


Figure 2.5: Prior and Lopez's RTO framework scheme (PRIOR and LOPEZ, 1999).

In Prior and Lopez's application, the data reconciliation ran with a 10-minute period associated with a strategy to compensate for dynamic effects and the optimizer ran with a 10-minute period too. The authors claimed that a more frequent approach overcomes the classic RTO as seeking "90 % of the benefits in 50 % of the

time would be better than achieving the optimum". As a result, the frequent RTO improved the plant throughput by 3-4 %, which resulted in a benefit of £1.5 million per year. Unfortunately, the proposed methodology to compensate for the plant dynamics was not disclosed in detail, which makes the proposition unpractical and prevents its replicability. The authors indicated that there was a proprietary interest in it, which would be the main significance of their work.

SEQUEIRA *et al.* (2002) and SEQUEIRA *et al.* (2004), also in an attempt to increase the RTO frequency, proposed a new framework that the authors called real-time evolution (RTE), as it does not perform the economic optimization step. In replacement to the optimization step, an approach based on sensitivity analysis runs frequently in order to continuously improve the actual operating point in the direction of decreasing the objective function. However, steady-state detection was still kept for the data reconciliation and parameter estimation steps. The authors claimed that the RTE outperforms RTO in the presence of drifting disturbances, but they considered the disturbances to be perfectly measured, which is rarely the case. Later, HUANG *et al.* (2011) claimed that RTE has no way of dealing with local optimum trap and proposed an "Improved" RTE by performing an economic optimization using a global solver, such as particle swarm optimization (PSO) or genetic algorithm (GA). However, this methodology change mischaracterized the RTE, whose main aspect is not to run a proper optimization, and ends up resembling more the approach proposed by BESL *et al.* (1998) than the actual RTE.

DARBY *et al.* (2011) reported that ARISTA *et al.* (2006) deployed an RTO in an Ultra-low Sulfur Diesel plant without the steady-state detection step. The authors proposed the use of the empirical Hammerstein–Wiener dynamic model to predict the process steady-state to be used in the data reconciliation and parameter estimation step. Unfortunately, no more details were found available. Despite the little information about the work, the idea behind it is very clear. An extra processing step is included between acquiring the plant measurement and the model adaption stage to translate the dynamic measurement on its static point once the plant settles. This translation would be done via a data-driven model, which would be restricted to the domain where the model was trained and validated. Therefore, this kind of technique would remove the extrapolation capability of a rigorous-mechanistic-model-based RTO.

2.6.1.2 HRTO or ROPA

VALLURU *et al.* (2015) proposed a completely separated control and optimization framework for processes with slow dynamics or frequent disturbances/parameters drifting without the steady-state detection step. The main idea

is to use a common Bayesian dynamic state and parameter estimator, e.g. EKF, for the economic optimization and an adaptive Nonlinear Model Predictive Control (NMPC). Another common aspect between the proposed layers is the use of the same dynamic model in the controller and in the observer and its steady form in economic optimization. Therefore, no model compatibility issues arise from the separate layers. In addition, the authors proposed a variable economic optimization frequency based on a pre-defined threshold for each disturbance or parameter being estimated by the EKF. Figure 2.6 illustrates the RTO framework proposed by VALLURU *et al.* (2015). The authors applied the proposed methodology in a Reactive Distillation process and reported significant operational improvements compared to the classic RTO scheme, especially during the transient periods.

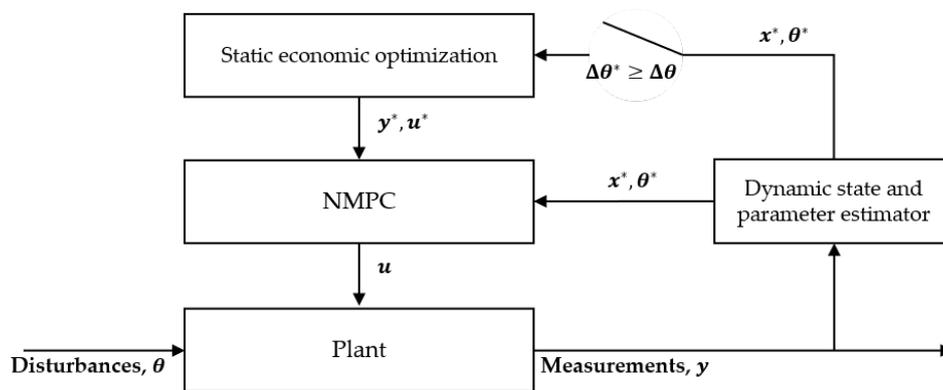


Figure 2.6: Valluru et al.’s RTO framework scheme (VALLURU *et al.*, 2015).

In 2018, two independent works were published almost simultaneously, both unaware of the work of VALLURU *et al.* (2015), claiming that they had proposed an RTO methodology that can use transient data, although both methodologies were identical to each other and very similar to Valluru et al.’s methodology. MATIAS and LE ROUX (2018) called it ROPA and KRISHNAMOORTHY *et al.* (2018b) called it HRTO. Despite none of them having actually been the first to propose the methodology, they have the merit of bringing more attention to the methodology and some enlightening discussions over the topic.¹

MATIAS and LE ROUX (2018) evaluated ROPA by applying it in a model of the Williams-Otto reactor and comparing it with the classic RTO approach. The authors emphasized the elimination of the steady-state waiting time and more freedom on the possibility of running the optimization at any desired time instant as the main benefits of ROPA. They also discussed that running RTO using transient data results in a continuous improvement of the objective function as

¹Throughout the text of the present thesis, the reader will notice several references to ROPA and HRTO, in order to prevent confusion it is important to clarify that the use of these acronyms refers in fact to the same optimization strategy where the parameter estimation is carried out in a dynamic framework followed by a static economic optimization.

the estimated parameters gradually converge to their expected values and, therefore, there is no difference between the final steady-state optimum reached by the classic RTO and by ROPA. Nonetheless, the authors also highlight that ROPA can destabilize the plant if the algorithm is not well-tuned, which is one important disadvantage of the methodology. The authors argue that the EKF tuning activity cannot be neglected and point to the execution frequency, optimal decision filtering, and EKF parameters as the main factors related to stability.

KRISHNAMOORTHY *et al.* (2018b) evaluated the ROPA algorithm in a simulated oil and gas production network and compared it to the classic RTO and to a DRTO approach. The main conclusion of the work was that ROPA is capable of presenting a similar economic performance to DRTO with a computational time similar to the classic RTO. The authors highlighted that there is no clear understanding of when to use static or dynamic optimization, although some processes present a dynamic nature that imposes the use of dynamic optimization, e.g. batch processes, cyclic operations, frequent grade changes, start-up, shut-down, etc. However, a dynamic problem can be considerably larger than its static counterpart because of the inclusion of the time dimension. For instance, in their case study, the dynamic NLP presented 139 times more decision variables than the static NLP. If a large-scale system were considered, this would be an even more dramatic issue, that could generate numeric issues and computational delays for the DRTO approach, resulting in performance degradation and possible closed-loop instabilities. KRISHNAMOORTHY *et al.* (2018b) also discussed other possibilities for the dynamic estimation methods, such as filtered bias update or implicit dynamic feedback (IDF) for simple one-to-one measurements to parameter relation, other variants of Bayesian methods, such as the UKF or the particle filter, or even more complex optimization-based methods such as the MHE. However, they highlighted the benefits of using EKF, as it is simple to implement, has low computational cost compared to the others, and corresponds to the maximum likelihood estimator for uncorrelated Gaussian white noise. However, the authors have not mentioned the lack of a clear method for EKF tuning for large-scale systems. Finally, the authors also outlined some concerns about modeling structural uncertainty and proposed the use of a bias term in the dynamic observer to cope with unmodelled effects. Nevertheless, no further results and comparisons were provided to support the effectiveness of this method, which appears to be insufficient to cope with the *model-adequacy condition* (CHACHUAT *et al.*, 2009; MARCHETTI *et al.*, 2016) considering that no input-affine correction was proposed to modify the original optimization problem.

In 2019, VALLURU and PATWARDHAN (2019) published a new research article that unveiled their work on RTO using transient data from 2015, showing

that they had proposed the methodology which would later be called ROPA or HRT0. The work does not bring novelty to the methodology proposed in 2015 but increases the discussion and the study cases. The authors thoroughly reviewed Bayesian methods that can be used in the parameter estimation stage with special attention to systems of differential-algebraic equations (DAEs). Moreover, three simulation case studies were carried out in the Williams-Otto reactor, a CSTR with input multiplicity and in an ideal reactive distillation column. The results confirmed that the proposed HRT0 coupled with adaptive NMPC can maintain the system at its economic optimum even under significant drifting disturbances/parameters.

SHAMAKI and ODLOAK (2020) tested the HRT0 methodology with a well-established zone-control infinite-horizon MPC proposed by GONZÁLEZ and ODLOAK (2009) in a simulated gas-lift system. The authors considered an EKF for estimating the gas-to-oil ratio of the wells and the HRT0 approach showed great synergy with the proposed control strategy, keeping the system in its optimal condition even after uncertain parameters step changes.

SANTOS *et al.* (2021) proposed the use of an Unscented Kalman Filter (UKF) and considered the objective function of the economic optimization as one of the controlled variables in an adaptive linear MPC approach and applied the proposed methodology to the Williams-Otto reactor benchmark. The controller objective was to keep the output variables within a desirable zone while tracking a setpoint for the economic objective function, in which the setpoint comes from the HRT0. The authors showed that the proposed adaptation strategy considerably reduces the model mismatch across the control and optimization layers. They claim that their approach can deal with structural uncertainty, but there is no evidence that this issue is tackled since no information about the plant gradients or mismatch modeling is used in the algorithm.

CURVELO *et al.* (2021) further investigated the use of HRT0 in a wide range of dynamic behaviors. The authors confirmed that the HRT0 strategy can in fact approach the economic benefits of the DRTO for most dynamic behaviors tested. However, special attention must be given to systems with considerably high dead times. In this case, the authors reported the arising of oscillatory modes that can even destabilize the system depending on the mismatch between the model and plant dead time. They proposed that a compensation methodology could be used to overcome this issue, such as a Smith predictor, but no further results were presented to support this claim. To this date, no research work has further explored ways to overcome the presence of dead time plant-model mismatch in HRT0.

MATIAS *et al.* (2022) applied the ROPA strategy in a real experimental rig and were the first authors to confirm the previous theoretical results in a true experi-

mental system. The authors implemented the classic RTO, ROPA, and DRTO approaches to the same experimental rig. They confirmed the close DRTO economic benefit and provided some guidelines for the practical implementation of ROPA.

The common bottleneck of all the previous works related to the HRTO or ROPA approaches is that all of them consider the availability of a rigorous dynamic process model, which is one of the main drawbacks of the HRTO since, in most real RTO applications, this assumption is not true. MATIAS and LE ROUX (2020) were the first to address this issue by proposing an asynchronous ROPA, where the problem is decomposed into static and dynamic sections and dynamic observers that rely on the rigorous dynamic model are used for the dynamic sections and steady-state detection is used for the static section. Based on this, a combination of steady-state and dynamic estimators is used asynchronously. The approach proposed by MATIAS and LE ROUX (2020) is more realistic to enable the application of ROPA in a plant-wide optimization scope. However, the availability of rigorous dynamic models is still mandatory for the process sections where the dynamic estimation takes place. Moreover, even though the authors have presented some guidelines, the activity of separating the whole plant into sections based on its dynamic behavior is not straightforward.

2.6.2 Problem adaptation schemes

In the context of MA, in which several iterations are required to achieve the real plant optimum, it is a significant improvement to use the transient measurements in order to speed up the optimization procedure.

FRANÇOIS and BONVIN (2014) highlight that the use of transient measurement in the MA scheme is a way to benefit from the two main advantages of the implicit and explicit RTO methods, that is fast convergence and explicitly handling with nonlinear constraints, respectively.

The main challenge in this context is how to estimate accurate steady-state gradients from transient data. Some authors proposed different methodologies in order to tackle this attempt, even outside the context of MA.

2.6.2.1 Seminal works

BAMBERGER and ISERMANN (1978) proposed a framework of identification and optimization in an implicit approach. The authors proposed the use of a nonlinear Hammerstein structure that is affine in the parameters to be estimated and nonlinear in the inputs, more specifically they chose a second-order input dependence, composed by n_y MISO models. The identification procedure was done by imposing a pseudo-random binary sequence (PRBS) perturbation to the plant

and a recursive algorithm based on least squared regression was used to estimate the parameters of the Hammerstein model. With the identified model at hand, the gradients and Hessian were calculated analytically from the model and used in an implicit optimization algorithm based on Newton's method. The authors tested the approach in a simulation case and an actual cooling water pilot plant. They reported a fast convergence to the plant optimum, without oscillatory behavior around it, and disturbances were successfully rejected. The main disadvantage of this approach is the requirement of imposing a PRBS perturbation to the plant, which is not well received by plant operators and managers. LEE and LEE (1985) proposed a cascade control and optimization of a fixed-bed reactor based on the methodology proposed by BAMBERGER and ISERMANN (1978). GOLDEN and YDSTIE (1989) extended the methodology proposed by BAMBERGER and ISERMANN (1978) in an adaptive extremum control combining a rigorous steady-state process model with the identification of a second-order Hammerstein model, including a forgetting factor in the regression scheme.

ZHANG and ROBERTS (1990) proposed a similar strategy to an ISOPE approach. A process identification during the transient stages was proposed based on a linear autoregressive exogenous (ARX) model structure. This linear ARX model can be interpreted as a special case of the Hammerstein model proposed by BAMBERGER and ISERMANN (1978), in which the model is affine in the parameters and the inputs. A Recursive Least Square (RLS) with a forgetting factor was proposed with a bootstrap correction to avoid biased estimation when noise is correlated (ROWE, 1970). Again, PRBS perturbation was imposed on the plant to guarantee accurate gradient estimation, which is the main drawback of the proposition.

2.6.2.2 Fast RTO or fast MA

In the context of MA, FRANÇOIS and BONVIN (2014) were the first to propose the use of transient measurements to speed up convergence. The strategy is frequently called "fast RTO" in recent literature (DE AVILA FERREIRA *et al.*, 2019a,b,c; MARCHETTI *et al.*, 2016), but the name "fast MA" would be more accurate to define the approach. The main idea is to estimate the steady-state modifier using the transient data aiming to achieve the plant's true optimum in a single settling time. The MA problem using transient data in a j^{th} iteration during the transient phase is:

$$\begin{aligned} \mathbf{u}_{j+1}^* = \arg \min_{\mathbf{u}} \quad & \mathcal{J}_m(\mathbf{u}) := J(\mathbf{u}) + (\hat{\boldsymbol{\lambda}}_j^J)^T(\mathbf{u} - \mathbf{u}_j) + \hat{\boldsymbol{\epsilon}}_j^J \\ \text{s.t.} \quad & G_i := g_i(\mathbf{u}) + (\hat{\boldsymbol{\lambda}}_j^{g_i})^T(\mathbf{u} - \mathbf{u}_j) + \hat{\boldsymbol{\epsilon}}_j^{g_i} \leq 0, \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \end{aligned} \quad (2.90)$$

with:

$$\widehat{\boldsymbol{\varepsilon}}_j^J = \widehat{J}_p(\mathbf{u}_j) - J(\mathbf{u}_j) \quad (2.91a)$$

$$\widehat{\boldsymbol{\varepsilon}}_j^{g_i} = \widehat{g}_{p,i}(\mathbf{u}_j) - g_i(\mathbf{u}_j), \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.91b)$$

$$\widehat{\boldsymbol{\lambda}}_j^J = \widehat{\nabla_{\mathbf{u}} J}_p(\mathbf{u}_j) - \nabla_{\mathbf{u}} J(\mathbf{u}_j) \quad (2.91c)$$

$$\widehat{\boldsymbol{\lambda}}_j^{g_i} = \widehat{\nabla_{\mathbf{u}} g}_{p,i}(\mathbf{u}_j) - \nabla_{\mathbf{u}} g_i(\mathbf{u}_j), \quad \forall i \in \mathbb{Z}_{\leq n_g}^+ \quad (2.91d)$$

in which the superscript ($\widehat{\cdot}$) represent an estimated steady-state value correspondent to \mathbf{u}_j . Note that in this section, the transient MA iteration is referred to with the index j . Again, the use of exponential filtering in inputs and modifiers was recommended. The authors showed that if the plant reaches a steady state and the values of \widehat{J}_p , $\widehat{\mathbf{G}}_p$, $\widehat{\nabla_{\mathbf{u}} J}_p$ and $\widehat{\nabla_{\mathbf{u}} \mathbf{G}}_p$ converge to their true values, then the plant will satisfy its NCO conditions and the steady-state will be a KKT point.

To calculate the zeroth-order modifiers, the authors took the difference between the actual plant measurement at time t_j and the modeled steady-state:

$$\widehat{J}_p(\mathbf{u}_j) = J_p(t_j) \quad (2.92a)$$

$$\widehat{\mathbf{G}}_p(\mathbf{u}_j) = \mathbf{G}_p(t_j) \quad (2.92b)$$

To calculate the first-order modifiers, the authors compared two gradient estimation methods: the Multiple-Unit (MU) and the Neighboring Extremals (NE). The MU method is very limited, it assumes that there are $n_u + 1$ identical units of the plant, the main one is used to proceed with optimization, and the others are perturbed to calculate the gradient by finite differences, so it is very sensitive to noise and its application to large industrial plant would be very difficult. Moreover, although the assumption of multiple identical plants is fair in the fields of electricity and electronics, it is hardly observed in the process industry. The NE approach is more interesting, it relies on variational analysis around the nominal operating point:

$$\begin{aligned} \widehat{\nabla_{\mathbf{u}} J}_p(\mathbf{u}_j^*) &= \nabla_{\mathbf{u}} J(\mathbf{u}_0^*, \boldsymbol{\theta}) + \nabla_{\mathbf{u}\boldsymbol{\theta}}^2 J(\nabla_{\boldsymbol{\theta}} \mathbf{y})^\dagger \delta \mathbf{y}_p(t_j) + \\ &\quad [\nabla_{\mathbf{u}\mathbf{u}}^2 \mathbf{J} - \nabla_{\mathbf{u}\boldsymbol{\theta}}^2 \mathbf{J}(\nabla_{\boldsymbol{\theta}} \mathbf{y})^\dagger \nabla_{\mathbf{u}} \mathbf{y}^T] \delta \mathbf{u}^*(t_j) \end{aligned} \quad (2.93a)$$

$$\begin{aligned} \widehat{\nabla_{\mathbf{u}} \mathbf{G}}_p(\mathbf{u}_j^*) &= \nabla_{\mathbf{u}} \mathbf{G}(\mathbf{u}_0^*, \boldsymbol{\theta}) + \nabla_{\mathbf{u}\boldsymbol{\theta}}^2 \mathbf{G}(\nabla_{\boldsymbol{\theta}} \mathbf{y})^\dagger \delta \mathbf{y}_p(t_j) + \\ &\quad [\nabla_{\mathbf{u}\mathbf{u}}^2 \mathbf{G} - \nabla_{\mathbf{u}\boldsymbol{\theta}}^2 \mathbf{G}(\nabla_{\boldsymbol{\theta}} \mathbf{y})^\dagger \nabla_{\mathbf{u}} \mathbf{y}^T] \delta \mathbf{u}^*(t_j) \end{aligned} \quad (2.93b)$$

in which $\delta \mathbf{y}_p(t_j) = \mathbf{y}_p(t_j) - \mathbf{y}_0^*$ and $\delta \mathbf{u}(t_j) = \mathbf{u}(t_j) - \mathbf{u}_0^*$ are the deviation of the current output and input from their nominal operating point. The superscript $(\cdot)^\dagger$ represents the Moore-Penrose pseudo-inverse operation. The authors showed that this approach is able to converge to the plant optimum within the settling time

of the plant.

Later, DE AVILA FERREIRA *et al.* (2017) expanded the method proposed by FRANÇOIS and BONVIN (2014). The authors observed that, with the proposed NE approach, when a steady-state gradient estimate is inaccurate, it could lead to oscillatory behavior or even prevent convergence. They proposed the use of a dynamic model to process the actual transient measurement and predict a corresponding steady state. An important requirement is that this transient model has to predict the same output values as the static model for any input value. In other words, the dynamic model used to predict the steady-state conditions must be compatible with the static model used in the optimization. The authors proposed the following correction instead of using directly the current values of the cost and constraints as proposed in Equation 2.92:

$$\widehat{J}_p(\mathbf{u}_j) = J^s(\mathbf{u}_j, \boldsymbol{\theta}) + [J_p(t_j) - J^{dyn}(t_j)] \quad (2.94a)$$

$$\widehat{\mathbf{G}}_p(\mathbf{u}_j) = \mathbf{G}^s(\mathbf{u}_j, \boldsymbol{\theta}) + [\mathbf{G}_p(t_j) - \mathbf{G}^{dyn}(t_j)] \quad (2.94b)$$

$$\widehat{\mathbf{y}}_p(\mathbf{u}_j) = \mathbf{y}^s(\mathbf{u}_j, \boldsymbol{\theta}) + [\mathbf{y}_p(t_j) - \mathbf{y}^{dyn}(t_j)] \quad (2.94c)$$

here, the superscripts $(\cdot)^s$ and $(\cdot)^{dyn}$ are used only to clarify the use of the static model or the dynamic model, respectively. The authors showed that these corrections vanish upon convergence and the plant KKT point holds. They reported the ability of fast MA to converge within the settling time of the plant, the absence of oscillatory behavior while decreasing RTO convergence period, in contrast to the method proposed by FRANÇOIS and BONVIN (2014), a rather insensitive response to filter gains and a satisfactory performance in a dynamic system that present non-minimum phase behavior. However, this extension included the requirement of having a compatible dynamic model, which highly hinders the applicability of such an approach.

A simpler version of this methodology was successfully applied to real solid-oxide fuel-cell systems (DE AVILA FERREIRA *et al.*, 2019a,b,c). The authors showed that for their particular problem, gradient errors in cost and constraints functions had no effect on the set of active constraints, so a simple fast CA could be implemented using a dynamic model to correct the modifiers during transient phases, and no gradient estimation technique was required. They showed that even under significant plant-model mismatch, the fast CA approach was able to quickly converge to plant optimum since it was an intersection of some active constraints.

RODRÍGUEZ-BLANCO *et al.* (2017) pointed out several drawbacks of the NE approach proposed by FRANÇOIS and BONVIN (2014), the most critical is that the method only works well when the source of model-plant mismatch is mainly

parametric, which contradicts the main motivation of using MA approaches, that is the presence of structural plant-model mismatch. In addition, it is required to know which are the uncertain parameters and there must be as many measurements as uncertain parameters.

SPEAKMAN and FRANÇOIS (2020) extended the work of FRANÇOIS and BONVIN (2014) for cases where the model's degrees of freedom are not the same as the plant's. Their methodology enables the gradient estimation of a controlled plant by using an open-loop model. The authors applied the method in a simulated CSTR and showed its ability to rapidly converge to a near-optimal region. However, the method also relied on NE gradient estimation, therefore the previously mentioned limitations hold.

RODRÍGUEZ-BLANCO *et al.* (2017) proposed a methodology based on the assumption that the variation of the process cost function can be approximated by a quadratic Taylor expansion and, then, a recursive extended least squares (RELS) regression with forgetting factor α was applied.

$$\Delta J_j \approx \Delta \hat{J}_j := \boldsymbol{\varphi}_{j-1}^T \hat{\boldsymbol{\Theta}}_{j-1} = \Delta \mathbf{u}_{j-1}^T \nabla_{\mathbf{u}_{j-1}} J + 1/2 \Delta \mathbf{u}_{j-1}^T \nabla_{\mathbf{u}_{j-1}}^2 J \Delta \mathbf{u}_{j-1} \quad (2.95)$$

in which $\Delta \mathbf{u}_{j-1} = \mathbf{u}_{j-1} - \mathbf{u}_{j-2}$, $\boldsymbol{\varphi}_{j-1}^T = [\Delta \mathbf{u}_{j-1}^T, 1/2 \Delta \mathbf{u}_{j-1}]$ and $\hat{\boldsymbol{\Theta}}_{j-1}^T = [\nabla_{\mathbf{u}_{j-1}} J, \nabla_{\mathbf{u}_{j-1}}^2 J \Delta \mathbf{u}_{j-1}]$.

The gradients are estimated by the following algorithm:

$$\boldsymbol{\Sigma}_0 = \frac{1}{\alpha} I \quad (2.96)$$

$$\begin{aligned} (\boldsymbol{\Sigma}_j)^{-1} &= \frac{1}{\alpha} (\boldsymbol{\Sigma}_{j-1})^{-1} - \frac{1}{\alpha^2} (\boldsymbol{\Sigma}_{j-1})^{-1} \boldsymbol{\varphi}_{j-1} \\ &\quad \left(1 + \frac{1}{\alpha} \boldsymbol{\varphi}_{j-1}^T (\boldsymbol{\Sigma}_{j-1})^{-1} \boldsymbol{\varphi}_{j-1} \right)^{-1} \boldsymbol{\varphi}_{j-1}^T (\boldsymbol{\Sigma}_{j-1})^{-1} \end{aligned} \quad (2.97)$$

$$\hat{\boldsymbol{\Theta}}_j^T = \hat{\boldsymbol{\Theta}}_{j-1}^T + \frac{1}{\alpha} (\boldsymbol{\Sigma}_{j-1})^{-1} \boldsymbol{\varphi}_{j-1} \left(1 + \frac{1}{\alpha} \boldsymbol{\varphi}_{j-1}^T (\boldsymbol{\Sigma}_{j-1})^{-1} \boldsymbol{\varphi}_{j-1} \right)^{-1} (\Delta J_j - \Delta \hat{J}_j) \quad (2.98)$$

in which $\boldsymbol{\Sigma}_j$ is the covariance matrix of the estimated error at the j^{th} iteration.

In order to avoid inaccuracy due to lack of excitation caused by passive identification, the authors added the dual constraint used in the dual ISOPE algorithm, $\varphi_{\kappa_j}(\mathbf{u}) - 1 \leq 0$, to the modified problem that is solved every sampling time at the transient phase. However, this approach does not fully characterize the duality due to the lack of the constraint that ensures that \mathbf{u} does not belong to the hyperplane \mathcal{H}_j , so it is not guaranteed that \mathbf{U}_j is regular. The authors applied the proposed scheme in an industrial depropanizer distillation column, the rigorous model was used as a plant and a reduced model was generated to be the

MA model, so the plant-model mismatch was simulated based on the difference between the rigorous and the reduced models. The reduced model was produced by reducing the number of trays to 3 and using a global efficiency approach. The number of trays of the original model was not disclosed by the authors, but they mention that the total number of equations was reduced from 1076 to 39. They compared the proposed scheme with the static dual MA (BRDYŚ and TATJEWSKI, 1994), the static nested MA (NAVIA *et al.*, 2015), and the fast MA (FRANÇOIS and BONVIN, 2014). They showed that the RELS approach was able to drive the real plant to its optimum 8 times faster than the static MA approaches and that the transient MA proposed by FRANÇOIS and BONVIN (2014) failed to reach the plant optimum due to severe plant-model mismatch.

2.6.2.3 MAWQA using transient measurements

Also addressing the problem of long waits for steady-state conditions and trying to avoid the need for extra plant excitation to accurately estimate gradients, GAO and ENGELL (2016) proposed the identification of linear ARX structures during the transient to predict the next steady-state point. The order of the linear ARX is determined according to the rigorous model of the plant. The authors proposed the use of additional conditions to approve the predicted steady-state point, i.e. mass and energy balances. If approved, the point predicted by the linear ARX model is used for static gradient estimation in an MAWQA scheme. Otherwise, the point is discarded, the data window is moved and a new ARX model is identified. The authors argued that, with the evolution of the transient data, the process will eventually lie in a linear region that can be satisfactorily approximated by a linear ARX model. The authors tested the methodology in the same CSTR case study used by FRANÇOIS and BONVIN (2014) and reported a result slightly slower, with approximately two settling times to converge to plant optimum. This delay is explained by the time required to approve the linear ARX in a conservative condition. The proposed methodology presents the advantage of working well with model structural uncertainty and unknown uncertain parameters, despite the NE methodology.

CADAVID *et al.* (2017) extended the methodology proposed by GAO and ENGELL (2016) to a more generalized framework based on the identification of a nonlinear ARX (NARX) during the transient phase to estimate the steady-state point. This estimated point was then used to estimate gradients in an MAWQA scheme. The authors proposed the use of a structure selection strategy based on a modified version of the bootstrap algorithm in the initial 50 sampling times, then the model parameters were estimated in a moving-window scheme. Similarly to GAO and ENGELL (2016), a conservative condition was imposed to approve the

estimated steady-state point. The authors tested the proposed methodology in the same CSTR case study then GAO and ENGELL (2016) and FRANÇOIS and BONVIN (2014) but in a different scenario. They tested the linear approach, the proposed nonlinear approach, a mixed approach, and the MAWQA without the use of transient data. All of them were able to drive the plant to its real optimum with 2.75τ , 2.51τ , 1.6τ , and 5.6τ , in which τ is the settling time, respectively. The approach showed the enhanced performance of the nonlinear approach over the linear. However, the most interesting result is that a mixed approach was able to be even better than the nonlinear one. To justify that effect, the authors argued that there is a trade-off between fast convergence and accurate convergence. When the system is far from the true optimum, the fast estimate obtained from the NARX structure is desirable and when it is near the optimum, the accurate estimate from the ARX structure is desirable. Moreover, the authors show that even under the detrimental effect of measurement noise, the use of transient measurement is still able to reduce convergence time with similar profit loss compared to static MA.

2.6.3 Optimizing control approaches

Several approaches integrate the optimization goals in the control layer, generating a single control and optimization layer. DARBY *et al.* (2011) discussed the benefit of keeping the two layers separately in practical applications, it eases the tuning efforts, maintenance, and troubleshooting. Nevertheless, it is worthwhile mentioning some of the current methodologies.

DE GOUVÊA and ODLOAK (1998) was the seminal work that proposed the inclusion of economic goals to the model predictive control objective function. This approach was later called EMPC. Later, DE SOUZA *et al.* (2010) proposed the inclusion of an additional tracking term to the MPC objective function to drive the gradient of the economic function to zero. DEMUNER *et al.* (2022) proposed a one-layer nonlinear EMPC framework in the absence of first-principle models, in which the static and dynamic terms are decomposed in a model structure that combines a *a priori* identified Gaussian Process for the system's steady states and several MISO Hammerstein models for the system's dynamics. There is a vast literature on EMPC, but a review on this subject is beyond the scope of this thesis. For this topic, we refer to ELLIS *et al.* (2014) and ELLIS *et al.* (2017).

Several other control techniques aim to achieve the optimum of an objective function. ESC determines the optimal setpoints that minimize an objective function by imposing frequent perturbations on system inputs (DOCHAIN *et al.*, 2011). SOC aims to determine the linear combination of control variables that result in a minimum loss when kept constant (JÄSCHKE *et al.*, 2017; SKOGES-

TAD, 2000). The NCO tracking is another control technique that aims to drive the plant gradient to zero employing a Newton method (JÄSCHKE and SKOGESTAD, 2011). More recently, a feedback RTO was proposed, which is a method to control a model-based gradient of the economic function to a zero setpoint by using a PID feedback loop (KRISHNAMOORTHY *et al.*, 2019). The modifier adaptation technique was also formulated as a feedback control problem in MARCHETTI *et al.* (2020). Another interesting approach that has gained increasing interest in recent academic research is the use of Reinforcement Learning techniques for control and optimization purposes (FARIA *et al.*, 2024, 2022, 2023).

Although these feedback approaches seem simple and practical, none of them naturally handles constraints, except the EMPC that solves a constrained optimization problem. This fact includes a great challenge to handle changes in the active set of constraints and it is still an open topic in the literature.

The next sections dive into SOC and RL, which are the optimizing control approaches of most interest in the present thesis. For a thorough review of the other methods, we refer to KRISHNAMOORTHY and SKOGESTAD (2022).

2.6.3.1 Self-optimizing Control

The motivation for the search for a self-optimizing control structure dates back to the 80s when MORARI *et al.* (1980) introduced the search for a function of the process variables that, when kept constant, would lead to optimal operation.

$$\mathbf{c} = h(\mathbf{y}) \tag{2.99}$$

However, this goal was then simplified by SKOGESTAD *et al.* (1998), which defines the self-optimizing control variables to be a linear combination of the measured variables:

$$\mathbf{c} = \mathbf{H}\mathbf{y} \tag{2.100}$$

and keeping the SOC variables, \mathbf{c} , constant at their optimal setpoints, \mathbf{c}_s , would lead to an acceptable loss in an economic index. Although the literature tends to focus mostly on economic objectives, this methodology can apply to any kind of goal, e.g. productivity, quality, environmental, etc. The loss function is defined as the gap between the optimal objective function and the actual objective function resulting from the process operation with a certain process control architecture. That is:

$$L_c := J(\mathbf{u}, \boldsymbol{\theta}) - J^*(\boldsymbol{\theta}^*) \tag{2.101}$$

Notice that this definition follows the optimization formulation presented in Equation 2.6, but here θ is used to express the process disturbances that also impact the process control.

The definition of a loss function and the introduction of an acceptable loss, first done by SKOGESTAD *et al.* (1998) and later formalized by SKOGESTAD (2000), represented a change of perspective by relaxing the goal of achieving optimality and enabled significant advances in the SOC field. Primarily, the active constraints are naturally the first SOC variables to be chosen as controlled variables, then with the remaining measured variables, one should carefully select how to select the remaining controlled variables. Therefore, later works on SOC focused on developing methods for the selection of the SOC variables by obtaining the H-matrix and how to deal with changes in the active set of constraints.

One of the first methods to be developed was the so-called brute force method (SKOGESTAD, 2000). The main idea of the method is to select the best CV out of the set of CV candidates by evaluating the maximum or average loss across sampled scenarios of all disturbances and noise realizations. However, this approach requires solving many large-scale non-convex optimization problems and can be intractable depending on the number of measurements.

Local methods were later developed by expanding the objective function in a second-order truncated Taylor series, and assuming the active set of constraints does not change during the operation (JÄSCHKE *et al.*, 2017). The local approximation is done based on the assumption that the SOC variables will perform satisfactorily around the nominal conditions neighborhood for most of the operation time. If this assumption does not hold for a particular application, local approaches shall not be considered. The constant active set of constraint assumption can be made by later exploring methods of dealing with changes in the constraints active set.

By applying the local method, the measurement model is linearized at the nominal condition:

$$\Delta \mathbf{y} = \mathbf{G}_u \Delta \mathbf{u} + \mathbf{G}_\theta \Delta \boldsymbol{\theta} + \mathbf{w}_y \quad (2.102)$$

in which, $\mathbf{G}_u = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{u}}\right)^T$ and $\mathbf{G}_\theta = \left(\frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}}\right)^T$ are the measurement Jacobian matrices with respect to \mathbf{u} and $\boldsymbol{\theta}$, respectively, evaluated at the nominal conditions. Here, the symbol Δ is used to represent the deviation from the nominal conditions, e.g. $\Delta \mathbf{y} := \mathbf{y} - \mathbf{y}_{nom}$, and $\mathbf{w}_y \in \mathbb{R}^{n_y}$ is the measurement noise vector.

It is then possible to come up with an explicit formulation for the loss function,

JÄSCHKE *et al.* (2017) provides the full demonstration:

$$L = \frac{1}{2} \left\| \mathbf{J}_{uu}^{1/2} (\mathbf{H}\mathbf{G}_u)^{-1} \mathbf{H}\mathbf{Y} \begin{bmatrix} \boldsymbol{\theta}' \\ \mathbf{w}' \end{bmatrix} \right\|_2^2 \quad (2.103)$$

in which, $\boldsymbol{\theta}' := \mathbf{W}_\theta^{-1} \Delta\boldsymbol{\theta}$ and $\mathbf{w}' := \mathbf{W}_{w_y}^{-1} \Delta\mathbf{w}$ are defined as scaled quantities based on the disturbances and noise magnitudes defined by matrices \mathbf{W}_θ and \mathbf{W}_{w_y} . $\mathbf{Y} = [\mathbf{F}\mathbf{W}_\theta \ \mathbf{W}_{w_y}]$, and the matrix $\mathbf{F} := (\frac{\partial \mathbf{y}^*}{\partial \boldsymbol{\theta}})^T$ is the sensitivity of the optimal measurements with the disturbances, which is computed by the following expression:

$$\mathbf{F} = \mathbf{G}_\theta - \mathbf{G}_u \mathbf{J}_{uu}^{-1} \mathbf{J}_{u\theta} \quad (2.104)$$

That stated, the exercise of finding the optimal SOC variables using the local approximation method, can be summarized by finding an implementable H-matrix that minimizes the loss function in Equation 2.103. However, such a formulation may lead to a non-convex problem with multiple solutions. Therefore, other approaches were developed to overcome this difficulty and compute the H-matrix. The most notable ones are:

- The **Null-space method** (ALSTAD and SKOGESTAD, 2007): this method assumes that the measurement can be neglected and that the number of measurements is greater or equal to the sum of numbers of inputs and disturbances ($n_y \geq n_u + n_\theta$). These assumptions allow to achieve zero loss provided that the H-matrix is selected in the left null space of \mathbf{F} , such that $\mathbf{H}\mathbf{F} = \mathbf{0}$, and $\mathbf{H}\mathbf{G}_u$ is non-singular. The H-matrix is calculated as:

$$\mathbf{H} = [\mathbf{J}_{uu} \ \mathbf{J}_{u\theta}] \tilde{\mathbf{G}}_u^\dagger \quad (2.105)$$

in which $\tilde{\mathbf{G}}_u := [\mathbf{G}_u \ \mathbf{G}_\theta]$. This method may result in uncontrollable SOC variables, in which their gains in relation to the inputs are zero. Furthermore, not considering the measurement noise makes this method not optimal in reality, which means that zero loss would not be achievable.

- The **Extended Null-space method** (ALSTAD *et al.*, 2009): the method considers $n_y < n_u + n_\theta$, and selects the H-matrix in a way that the priority is to reject disturbances, then the remaining measurements are used to minimize the effect of noise. The H-matrix is computed by:

$$\mathbf{H} = [\mathbf{J}_{uu} \ \mathbf{J}_{u\theta}] (\mathbf{W}_n^{-1} \tilde{\mathbf{G}}_u)^\dagger \mathbf{W}_n^{-1} \quad (2.106)$$

A problem with this method is the inability to provide an optimal balance between disturbance and noise rejection.

- The **Minimum Loss method**: ALSTAD *et al.* (2009) formulated a convex optimization problem to find the H-matrix that minimizes the average loss.

$$\begin{aligned} \min_H \quad & \|FY\|_F \\ \text{s.t.} \quad & HG_u = J_{uu}^{1/2} \end{aligned} \tag{2.107}$$

They found an explicit solution, which was later refined by YELCHURU and SKOGESTAD (2012):

$$H = (G_u)^T (YY^T)^{-1} \tag{2.108}$$

This result provides the best measurement combination locally by providing an optimal trade-off between disturbance and noise rejection.

- The **Taylor-made Branch and Bound** (CAO and KARIWALA, 2008; KARIWALA and CAO, 2009, 2010): this method is used to find the best measurement set rather than the measurement combination, that is, the H-matrix is composed of only zeros and ones, and only a single one by row. The method exploits the fact that removing a measurement variable cannot decrease the loss function to select or discard branches from the search. As a result, a very efficient method to screen the best of all possible subsets of measurement, without actually having to evaluate all of them, as it is done in the brute force methods.
- The **MIQP formulation**: also in the attempt to find the best measurement subset rather than a combination, YELCHURU and SKOGESTAD (2012) proposed to reformulate the Problem 2.107 as a mixed integer quadratic programming (MIQP) problem, and associate integer variables to control which measurement are active or not. A constraint is added to ensure that only one variable is selected per row in the H-matrix. However, the formulation adds a tuning parameter that is hard to find, which results in the requirement of solving the MIQP problem multiple times iteratively to make sure the solution found is optimal.

All the methods described above assume a constant active constraint set. However, in practice, the disturbances can cause a change in the active set of constraints, and a change in the optimal SOC variables by consequence. Some methods proposed for dealing with changes in the active set of constraints are:

- The **cascade control approach** (CAO, 2004): this approach proposes to set a cascade control approach, where the inner loop is responsible for the constraint control, and the outer loop is responsible for the self-optimizing control. A saturation is imposed on the outer loop to guarantee that the constraint is not violated. However, this approach can only be applied when there is a one-to-one mapping between constraints and CVs, and it results in loss whenever a constraint turns active.
- The **integrated approach** (HU *et al.*, 2012): this approach aims to find a single H-matrix that ensures all variables are within their bounds. It modifies the Minimum Loss method to include the consideration of the constraints:

$$\begin{aligned}
\min_H \quad & \frac{1}{2} \|\mathbf{J}_{uu}^{1/2} \mathbf{H} \mathbf{Y}\|_F^2 \\
\text{s.t.} \quad & \mathbf{H} \mathbf{G}_u = \mathbf{I} \\
& \|B_i\|_1 \leq 0, \forall i \in \mathbb{I}_{n_g}
\end{aligned} \tag{2.109}$$

where B_i , comes from the linearization of each constraint g_i , and some further manipulations. We refer to the authors for more details. There is no guarantee that this optimization problem will have a solution. If it has, the operation will result in loss when a constraint is activated. Also, the linear approximation of the constraints can generate important plant-model mismatch in a real plant operation, which could result in even more losses, and possibly prohibitive constraint violations.

- The **parametric programming approach** (MANUM and SKOGESTAD, 2012): the authors proposed to identify the active constraint regions by solving a parametric programming problem considering the disturbances as the parameters. Then, different optimal SOC variable configurations are designed for each of these regions. This involves approximating the steady-state optimization problem as a QP. Then, if the problem has a solution, an invariable SOC configuration is derived for each region. To switch between regions, the authors proposed to monitor a scalar descriptor function for all neighbors and current regions. The switch is done when the signs of the descriptor functions change. Another strategy for switching between regions is the use of max/min selectors (KRISHNAMOORTHY and SKOGESTAD, 2022). The parametric programming approach may not be applicable in practice. The number of regions may be too big, and the QP approximation can result in severe approximation errors that can pose great uncertainty on the resulting region boundaries. Also, there is no guarantee that the parametric programming solver will converge for every case. Another

commonly reported issue is the complexity of the approach, resulting in too many different and complex control configurations for each active constraint set.

For a deeper review of these methods, challenges, and developments in the field, we refer to JÄSCHKE *et al.* (2017). It is worth mentioning that, it has been shown that controlling the SOC variables is an indirect way of achieving the NCO when the H-matrix is chosen in the left null space of the sensitivity matrix of the optimal measurements with the disturbances (JÄSCHKE and SKOGESTAD, 2010). In this case, SOC and NCO tracking (FRANÇOIS *et al.*, 2005; KRISHNAMOORTHY *et al.*, 2019) may be seen as equivalents, although the pursuit of the economic function optimality has been relaxed, and it is not a direct goal of the main SOC methods.

2.6.3.2 Reinforcement Learning

Reinforcement Learning (RL) is a class of algorithms that autonomously learns how to realize a specific task via direct unsupervised interactions with an environment (SUTTON and BARTO, 2018b). The success of RL in many areas, such as the development of chess (SILVER *et al.*, 2018) and go (SILVER *et al.*, 2016) computer engines, the development of autonomous vehicles (KIM *et al.*, 2003), and robotics (ANDRYCHOWICZ *et al.*, 2020; HEESS *et al.*, 2017), has been induced the interest of the process control community in such topic. A more in-depth description of the RL problem is presented in Appendix D.

The first application of RL for chemical engineering problems dates from the early 1990s and can be found in HOSKINS and HIMMELBLAU (1992). In this pioneering work, the authors proposed a neural network (NN) based control, in which two different NN models are used to predict both the control performance measure and the policy. The method was applied to a non-isothermal CSTR example, and the results were compared with conventional PID control. Despite interesting results at that time, the number of publications remained low until the mid-2010s. During that period, artificial intelligence did not have a significant impact on chemical engineering since computing, storage and communication capacity, programming environments, and data quantity were insufficient to solve the complex problems that these techniques were meant to address (VENKATASUBRAMANIAN, 2018). However, the evolution of sensors and digital technologies in recent years has reduced these limitations, and the number of applications of RL in process control has increased exponentially ever since.

Recent review articles have discussed the potential of using RL-based methodologies for process control and optimization in industry (FARIA *et al.*, 2023; NIAN

et al., 2020; SHIN *et al.*, 2019a). In particular, in SHIN *et al.* (2019a), the authors thoroughly compare RL and MPC techniques regarding very important aspects to be considered for industrial applications such as online computational cost, state constraints enforcement, the controller algorithm convergence properties, the closed-loop stability properties, and adaptiveness capacity. The authors point out that RL techniques offer the following interesting characteristics against NMPC limitations: once trained, the computational cost for the computation of the control action is very low as it is obtained through the evaluation of a function (e.g. a neural network model), and RL techniques are naturally adaptive as the training of the agent may continue as new process data is collected. A comparison of RL-based control and NMPC subject to uncertainties has also been done in KORYAKOVSKIY *et al.* (2017). The results demonstrate that nonlinear model predictive control has advantages over reinforcement learning if uncertainties can be eliminated through the identification of the system parameters. Otherwise, there exists a break-even point after which model-free reinforcement learning performs better than nonlinear model predictive control with an inaccurate model. However, research work still has to be done regarding the convergence of the RL training algorithm, the closed-loop stability of the controller, and the enforcement of constraints.

For training, the actor-critic algorithm is the most popular approach for solving RL problems with continuous state and action spaces since it benefits from the advantages of the two other methodologies (i.e., value-based and policy gradient-based) (FARIA *et al.*, 2022). In this framework, the policy defining the decisions taken by the controller, and the value function assessing the policy are approximated by parameterized functions. For their great capacity of generalization, neural networks are the most used approximators at the present time.

Some of RL works related to PSE applications are enumerated below. In ANDERSON *et al.* (1997), RL is combined with PID for the control of a heating coil. In MARTINEZ (2000), RL is used for the optimization of batch processes, where the RL solution is used to systematically shrink the region of interest for the optimization. In SYAFIIE *et al.* (2008), RL techniques are used for pH and pollutant concentration control in wastewater plants. RL is combined with fuzzy logic for the control of a fermentation process in LI *et al.* (2011). RL is used to optimize a batch distillation process in WILSON (2012). In SHAH and GOPAL (2016), RL is used to control a non-isothermal CSTR by manipulating the PID controller tuning parameters. An RL-based controller of a conical tank system is proposed in RAMANATHAN *et al.* (2017), and the control of wastewater treatment plants using RL is tackled in HERNÁNDEZ-DEL OLMO *et al.* (2017).

Although there is a considerable number of papers linking RL with process

control, the integration with economic goals has not been as explored as setpoint tracking problems. In this context, POWELL *et al.* (2020) were the first to link RL in the context of RTO and propose an RL-RTO framework that is capable of pursuing steady-state economic optimization. The constraints are addressed by including extra penalty terms in the reward function, which is a straightforward idea but does not prevent infeasible solutions. The results showed that the RL-RTO was able to achieve half of the benefit of the classic RTO, which still shows the superiority of RTO, but an interesting potential for such RL-based methods.

Chapter 3

Development of a hybrid real-time optimization approach in the absence of a rigorous dynamic model via process identification of Hammerstein model structures

A version of this chapter was presented in the 31st European Symposium on Computer Aided Process Engineering (ESCAPE 31) and published in the Book series Computer Aided Chemical Engineering, Volume 50, 2021, Pages 259-265 (DELOU *et al.*, 2021b).

3.1 Introduction

This chapter addresses the HRTO requirement of having a rigorous dynamic process model. Three HRTO architectures that are able to perform in the absence of a rigorous dynamic model are developed and compared to two HRTO methodologies that use perfect dynamic models. To do so, two dynamic Hammerstein model structures are proposed to combine the information from an available nonlinear static model with a linear ARX to be identified from process data.

3.2 Hammerstein model structure

It is assumed that an adequate static model of the plant is available:

$$\begin{aligned} \mathbf{0} &= \mathcal{F}(\mathbf{x}^s, \mathbf{u}, \boldsymbol{\theta}) \\ \mathbf{y}^s &= \mathcal{H}(\mathbf{x}^s, \mathbf{u}, \boldsymbol{\theta}) \end{aligned} \quad (3.1)$$

in which, $\mathbf{x}^s \in \mathbb{R}^{n_x}$ are the state variables in stationary state, $\mathbf{y}^s \in \mathbb{R}^{n_y}$ are the measured variables in stationary state, $\mathbf{u} \in \mathbb{R}^{n_u}$ are the input variables and $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ are the vector of parameters and unmeasured disturbances on which uncertainty occurs. Also, the functions $\mathcal{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x}$ and $\mathcal{H} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_y}$ are the static mapping of the states and the outputs related to the inputs and parameters, respectively. For the sake of simplifying notation, this model will be represented as $\mathbf{y}^s = \mathbf{y}^s(\mathbf{u}, \boldsymbol{\theta})$.

A general Hammerstein structure consists of a static mapping of the inputs and parameters into the output variables and a linear dynamic model to make the transition from the current state to the mapped steady state. Considering a discrete linear dynamic model of the type of autoregressive with exogenous input (ARX) to approximate the plant dynamics, the Hammerstein model is given by:

$$\mathbf{y}_k^h = \mathbf{A}_k^h \mathbf{y}_{k-1}^h + \mathbf{B}_k^h \mathbf{y}_{k-1}^s(\mathbf{u}_{k-1}, \boldsymbol{\theta}_{k-1}) \quad (3.2)$$

in which, $\mathbf{y}_k^h \in \mathbb{R}^{n_y}$ is the vector of output variables provided by the Hammerstein model at instant k . In addition, the squared matrices \mathbf{A}_k^h and \mathbf{B}_k^h gather the dynamic parameters to be identified from process data during operation, but can also be kept constant.

Here, two models are proposed that exploit this structure, they are

1. model 1: $\mathbf{B}_k^h = (\mathbf{I} - \mathbf{A}_k^h)$;
2. model 2: \mathbf{A}_k^h and \mathbf{B}_k^h are independent.

It is noteworthy that model 1 presents the propriety to match the stationary conditions of the static model, and also it presents half of the dynamic parameters to be identified compared to model 2.

3.3 HRT0 architectures

The proposed HRT0 architectures based on the Hammerstein models presented in Section 3.2 are described below:

- **HRTO-HEKF-fixed:** The adjustable parameters are estimated by an Extended Kalman Filter (EKF) layer that considers the Hammerstein model as its internal dynamic model, named Hammerstein EKF (HEKF), wherein the dynamic matrices of the model are kept constant. The estimated parameters are then kept fixed in a subsequent static optimization layer;
- **HRTO-HEKF-adaptive:** The adjustable parameters and the dynamic matrices of the Hammerstein model are simultaneously estimated by the HEKF layer;
- **HRTO-HEKF-RELS:** The adjustable parameters are estimated by the HEKF layer but the dynamic matrices of the Hammerstein model are estimated by a Recursive Extended Least Squares (RELS) estimator with forgetting factor.

These approaches are compared with the following architectures:

- **HRTO-EKF:** the original HRTO, in which the true dynamic model of the process is known and the adjustable parameters are estimated by an EKF layer;
- **HRTO-LSE:** the original two-step RTO, in which the adjustable parameters are estimated by a nonlinear Least Squares Estimator (LSE) subjected to the static model. In this approach, no steady-state detection is performed, so the LSE runs regardless of the dynamic nature of the data.

In all approaches, it is considered the presence of a parametric plant-model mismatch, in which the vector of true plant parameters θ_p is unknown to the model.

3.3.1 Hammerstein EKF algorithm

The algorithm for the parameter estimation in the EKF framework is described in this section. The application for Hammerstein models in the HRTO-EKF-fixed gives the following augmented *a priori* state estimation:

$$\begin{bmatrix} \hat{\mathbf{y}}_k^{h-} \\ \hat{\boldsymbol{\theta}}_k^- \end{bmatrix} = \begin{bmatrix} \mathbf{A}_k^h \hat{\mathbf{y}}_{k-1}^{h+} + \mathbf{B}_k^h \mathbf{y}_{k-1}^s(\mathbf{u}_k, \hat{\boldsymbol{\theta}}_{k-1}^+) \\ \hat{\boldsymbol{\theta}}_{k-1}^+ \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{y,k} \\ \mathbf{w}_{\theta,k} \end{bmatrix} \quad (3.3)$$

in which, $\mathbf{w}_{y,k}$ and $\mathbf{w}_{\theta,k}$ are artificial zero mean noise, so that $\mathbf{w}_{y,k} \sim \mathcal{N}(0, \mathbf{Q}_y)$ and $\mathbf{w}_{\theta,k} \sim \mathcal{N}(0, \mathbf{Q}_\theta)$. Therefore, the process noise covariance matrix is augmented as $\tilde{\mathbf{Q}} = \text{diag}([\mathbf{Q}_y, \mathbf{Q}_\theta])$. In the HRTO-HEKF-adaptive the elements of the dynamic matrices are estimated simultaneously to the adjustable parameters, so the columns of these matrices are stacked up and concatenated with vector $\boldsymbol{\theta}$.

3.3.2 RELS algorithm

The algorithm of the RELS with forgetting factor α is very well presented in RODRÍGUEZ-BLANCO *et al.* (2017). Using their nomenclature, the application for the Hammerstein models are $\hat{\phi}_k = \mathbf{y}_k^h - \mathbf{y}_k^s$, $\varphi_{k-1}^T = \mathbf{y}_{k-1}^h - \mathbf{y}_{k-1}^s$ and $\hat{\beta}_k = (\mathbf{A}_k^h)^T$ for model 1 and $\hat{\phi}_k = (\mathbf{y}_k^h)^T$, $\varphi_{k-1}^T = [(\mathbf{y}_{k-1}^h)^T, \mathbf{y}_{k-1}^s]^T$ and $\hat{\beta}_k = [(\mathbf{A}_k^h)^T, (\mathbf{B}_k^h)^T]^T$ for model 2 and the initial covariance matrix of the prediction error is given by $\Sigma_0 = \mathbf{I}/\alpha$.

3.4 Case Study: The Williams-Otto Reactor

The Williams-Otto reactor with three reactions was exploited as a case study. It is a classical benchmark for real-time optimization studies, where several phenomena can be observed, such as inverse response and change of the gain sign. Its equations, parameter values, and notations can be found in FORBES and MARLIN (1996).

In this study, we consider w_C as an unmeasured variable, so the set of measured variables are $\mathbf{y} = [w_A, w_B, w_E, w_P, w_G]^T$, the degrees of freedom of the optimizer are $\mathbf{u} = [F_B, T_R]^T$ and the set of adjustable parameters were in fact two unmeasured disturbances $\boldsymbol{\theta} = [F_A, W]^T$. The kinetic parameters were considered fixed and perfectly known.

The economic objective function considered in both plant and model optimization is:

$$J = 1043.38w_P F + 20.92w_E F - 79.23F_A - 118.34F_B \quad (3.4)$$

To initialize the Hammerstein models, an offline identification problem was run around a nominal point, $\mathbf{u}_n = [2, 70]^T$ in [kg/s, °C] and $\boldsymbol{\theta}_n = [1.8275, 2105]^T$ in [kg/s, kg], distant from the nominal optimum point, $\mathbf{u}^* = [4.2, 85.9]^T$ in [kg/s, °C]. Finally, a scenario of parameter variation was designed to test the approaches. For the approach HRTO-HEKF-RELS, two values of the forgetting factor α were tested, 0.5 and 0.99.

In terms of the adopted Kalman filter tuning, $\mathbf{R} = 1 \cdot 10^{-4} \mathbf{I}_5$ for all approaches. For HRTO-EKF, HRTO-HEKF-fixed, and HRTO-HEKF-RELS, the $\tilde{\mathbf{Q}}$ -matrix was:

$$\tilde{\mathbf{Q}} = \begin{bmatrix} 1 \cdot 10^{-4} \mathbf{I}_5 & 0 & 0 \\ 0 & 5 \cdot 10^{-2} & 0 \\ 0 & 0 & 5 \cdot 10^4 \end{bmatrix} \quad (3.5)$$

For the HRTO-HEKF-adaptive approach, the \tilde{Q} -matrix was:

$$\tilde{Q} = \begin{bmatrix} 1 \cdot 10^{-4} \mathbf{I}_5 & 0 & 0 & 0 \\ 0 & 5 \cdot 10^{-2} & 0 & 0 \\ 0 & 0 & 5 \cdot 10^4 & 0 \\ 0 & 0 & 0 & 1 \cdot 10^{-8} \mathbf{I}_n \end{bmatrix} \quad (3.6)$$

in which, n comes from the number of dynamic parameters to be estimated for the Hammerstein model. For model 1, $n = n_y^2$, and for model 2, $n = 2n_y^2$, since model 2 has two dynamic matrices to be estimated, \mathbf{A}_k^h and \mathbf{B}_k^h , while model 1 has only one, \mathbf{A}_k^h . Therefore, $n = 25$ for Hammerstein model 1 and $n = 50$ for Hammerstein model 2 since the Williams-Otto model has $n_y = 5$ measured output variables.

A sampling time of $60s$ was considered in a simulation window of $N = 450$ sampling times in which a decreasing ramp was subjected to the reactor holdup, W , and a pulse disturbance was subjected to the feed flow rate of reactant A , F_A , as it is shown in Figure 3.1.

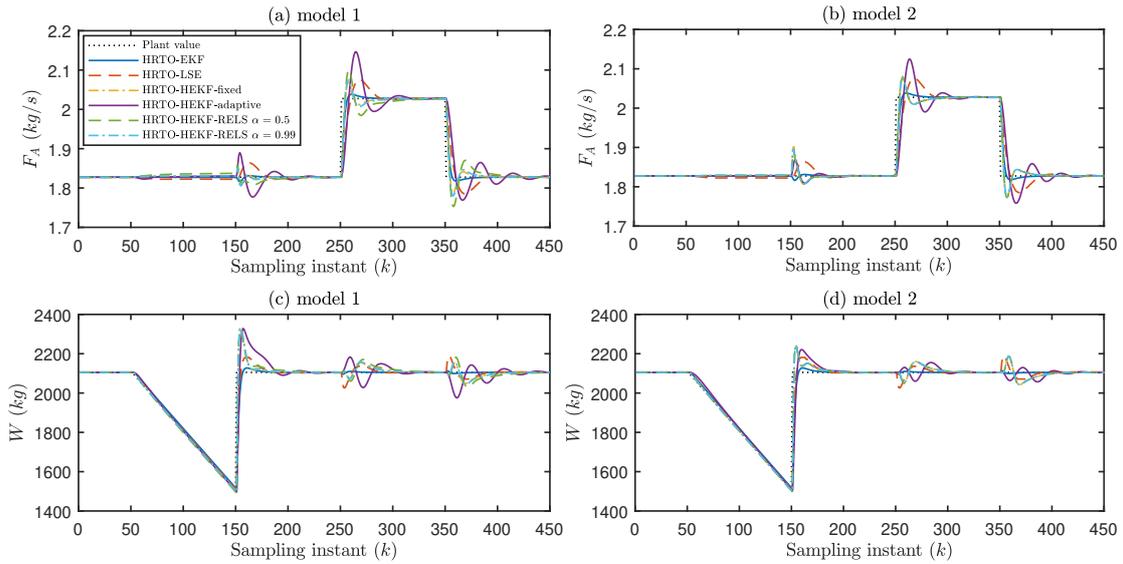


Figure 3.1: Performance of the different HRTO architecture over a scenario of parametric variation: (a) estimation of F_A with HEKF and RELS using model 1; (b) estimation of F_A with HEKF and RELS using model 2; (c) estimation of W with HEKF and RELS using model 1; (d) estimation of W with HEKF and RELS using model 2.

The decision variables can be visualized in Figure 3.2.

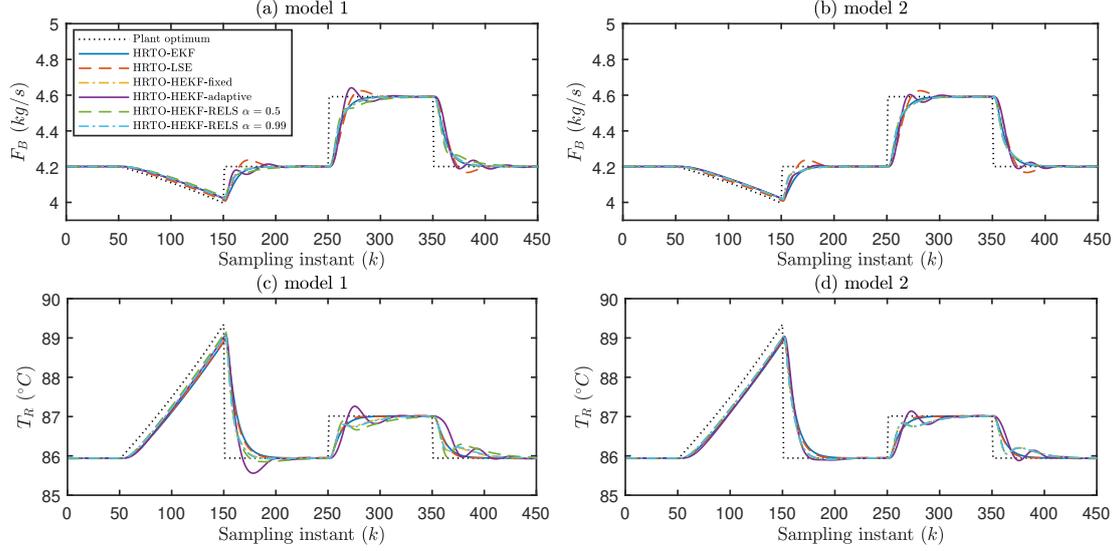


Figure 3.2: Performance of the different HRTO architecture over a scenario of parametric variation: (a) behavior of F_B with HEKF and RELS using model 1; (b) behavior of F_B with HEKF and RELS using model 2; (c) behavior of T_R with HEKF and RELS using model 1; (d) behavior of T_R with HEKF and RELS using model 2.

In order to compare all approaches in this scenario, a normalized Mean Squared Error (nMSE) for each variable is shown in Table 3.1.

Table 3.1: nMSE comparison between HRTO architectures

Variables	HRTO-EKF	HRTO-LSE	model 1				model 2			
			HRTO-HEKF		HRTO-HEKF-RELS		HRTO-HEKF		HRTO-HEKF-RELS	
			fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$	fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$
w_A	$4.6 \cdot 10^{-6}$	$3.4 \cdot 10^{-6}$	$5.8 \cdot 10^{-6}$	$8.4 \cdot 10^{-5}$	$6.0 \cdot 10^{-6}$	$5.8 \cdot 10^{-6}$	$5.9 \cdot 10^{-6}$	$6.6 \cdot 10^{-5}$	$5.9 \cdot 10^{-6}$	$5.9 \cdot 10^{-6}$
w_B	$3.8 \cdot 10^{-6}$	$4.1 \cdot 10^{-6}$	$4.8 \cdot 10^{-6}$	$3.2 \cdot 10^{-5}$	$4.7 \cdot 10^{-6}$	$4.8 \cdot 10^{-6}$	$4.7 \cdot 10^{-6}$	$3.0 \cdot 10^{-5}$	$4.7 \cdot 10^{-6}$	$4.7 \cdot 10^{-6}$
w_E	$2.4 \cdot 10^{-6}$	$3.4 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$	$1.7 \cdot 10^{-5}$	$3.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$	$1.6 \cdot 10^{-5}$	$3.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$
w_P	$6.9 \cdot 10^{-7}$	$2.5 \cdot 10^{-6}$	$8.5 \cdot 10^{-7}$	$4.6 \cdot 10^{-6}$	$8.5 \cdot 10^{-7}$	$8.5 \cdot 10^{-7}$	$9.1 \cdot 10^{-7}$	$5.1 \cdot 10^{-6}$	$9.1 \cdot 10^{-7}$	$9.1 \cdot 10^{-7}$
w_G	$2.3 \cdot 10^{-6}$	$2.4 \cdot 10^{-6}$	$3.3 \cdot 10^{-6}$	$1.8 \cdot 10^{-5}$	$3.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$	$2.4 \cdot 10^{-5}$	$3.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$
J	$4.1 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$4.8 \cdot 10^{-2}$	$2.8 \cdot 10^{-1}$	$5.4 \cdot 10^{-2}$	$4.9 \cdot 10^{-2}$	$5.5 \cdot 10^{-2}$	$3.0 \cdot 10^{-1}$	$5.4 \cdot 10^{-2}$	$5.4 \cdot 10^{-2}$
F_A	$1.7 \cdot 10^{-4}$	$3.4 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$	$6.0 \cdot 10^{-4}$	$2.7 \cdot 10^{-4}$	$1.9 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$	$4.9 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$
W	$5.9 \cdot 10^{-1}$	$7.0 \cdot 10^{-1}$	$7.5 \cdot 10^{-1}$	$14.0 \cdot 10^{-1}$	$8.5 \cdot 10^{-1}$	$7.6 \cdot 10^{-1}$	$6.3 \cdot 10^{-1}$	$9.8 \cdot 10^{-1}$	$6.3 \cdot 10^{-1}$	$6.2 \cdot 10^{-1}$

The nMSE for each variable i , using the true plant value as reference, is given by:

$$nMSE_i = \sum_{k=1}^N \frac{(\hat{y}_{i,k} - y_{i,k}^p)^2}{y_{i,k}^p} \quad (3.7)$$

in which, the hat emphasis ($\hat{\cdot}$) and the superscript (\cdot^p) represent the estimated value and the true plant value, respectively.

To compare each approach with the HRTO-EKF, which presents the overall best performance since it uses the same plant dynamic model structure, a relative

Mean Squared Error (rMSE) is introduced for each variable as follows:

$$rMSE_i = \frac{1}{nMSE_i^{\text{HRTO-EKF}}} \sum_{k=1}^N \frac{(\hat{y}_{i,k} - y_{i,k}^p)^2}{y_{i,k}^p} \quad (3.8)$$

Table 3.2 presents the rMSE for each HRTO approach.

Table 3.2: rMSE comparison between HRTO architectures

Variables	HRTO-LSE	model 1				model 2			
		HRTO-HEKF		HRTO-HEKF-RELS		HRTO-HEKF		HRTO-HEKF-RELS	
		fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$	fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$
w_A	0.75	1.26	18.53	1.31	1.27	1.29	14.39	1.29	1.29
w_B	1.08	1.26	8.36	1.25	1.25	1.23	7.81	1.23	1.23
w_E	1.41	1.33	7.08	1.32	1.33	1.35	6.71	1.35	1.35
w_P	3.74	1.24	6.98	1.25	1.24	1.33	7.49	1.33	1.33
w_G	1.07	1.43	7.79	1.4	1.42	1.43	10.46	1.43	1.43
J	4.32	1.19	6.99	1.33	1.21	1.35	7.25	1.32	1.33
F_A	2.06	1.08	3.58	1.63	1.14	1.25	2.94	1.2	1.22
W	1.19	1.27	2.4	1.44	1.3	1.07	1.67	1.07	1.06
Average	1.95	1.26	7.71	1.37	1.27	1.29	7.34	1.28	1.28

It is straightforward from Figure 3.1 that all approaches were able to estimate the true value of the parameters after a stabilization period, which confirms the potential of the Hammerstein approaches to substitute the rigorous dynamic model in the architecture of the HRTO, opening up a vast range of possibilities of applications. Surprisingly, the HRTO-LSE presented a good performance in parameter estimation during transient regions. That suggests that HRTO could be applied directly to classic RTO structures, just by means of removing the steady-state detection stage and fastening up the running frequency of the optimization loop. However, this approach is the most costly among the other structures analyzed, as shown in Table 3.3, it is almost 11.5 times more costly than EKF and 23 times more costly than the proposed HEKF. Besides, this preliminary conclusion about the HRTO-LSE should be verified by testing with many other case studies. Tables 3.3, 3.3, and 3.3 show the average, minimum, and maximum consumption times for each stage in the simulation loop.

Table 3.3: Average consumption time for each stage in loop (ms)

Variables	HRTO-EKF	HRTO-LSE	model 1				model 2			
			HRTO-HEKF		HRTO-HEKF-RELS		HRTO-HEKF		HRTO-HEKF-RELS	
			fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$	fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$
Plant optimization	2.04	2.08	2.08	2.08	2.25	2.05	2.07	2.08	2.40	2.15
Estimation	1.94	22.94	0.74	0.98	0.95	0.86	0.69	1.03	0.98	0.82
Model optimization	11.92	12.81	11.95	12.03	12.95	12.03	11.99	12.09	13.69	12.23
Total simulation loop	16.95	39.31	15.89	16.35	17.35	16.07	15.84	16.49	18.35	16.31

Table 3.4: Minimum consumption time for each stage in loop (*ms*)

Variables	HRTO-EKF	HRTO-LSE	model 1				model 2			
			HRTO-HEKF		HRTO-HEKF-RELS		HRTO-HEKF		HRTO-HEKF-RELS	
			fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$	fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$
Plant optimization	$7 \cdot 10^{-4}$	$7 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$6 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$7 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
Estimation	1.66	12.58	0.6	0.76	0.66	0.66	0.61	0.96	0.66	0.67
Model optimization	10.78	11.62	10.83	11.63	10.44	10.77	10.67	13.81	10.59	10.67
Total simulation loop	13.43	26.27	12.48	13.69	12.19	12.5	12.33	16.22	12.28	12.31

Table 3.5: Maximum consumption time for each stage in loop (*ms*)

Variables	HRTO-EKF	HRTO-LSE	model 1				model 2			
			HRTO-HEKF		HRTO-HEKF-RELS		HRTO-HEKF		HRTO-HEKF-RELS	
			fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$	fixed	adaptive	$\alpha = 0.5$	$\alpha = 0.99$
Plant optimization	12.83	15.91	19.06	17.69	17.65	13.62	15.15	23.09	13.62	13.69
Estimation	27.8	308.11	10.69	34.27	88.93	14.14	4.08	15.01	5.58	4.21
Model optimization	14.87	21.89	20.91	23.92	18.32	20.26	18.18	25.07	19.9	20.6
Total simulation loop	44.15	329.9	40.63	59.97	110.17	34.03	31.29	46.24	29.31	52.43

In general, approaches using model 1 presented a greater adaptation capability over approaches using model 2, due to the smaller number of parameters to estimate. However, this capability does not reflect in a better parameter estimation necessarily, in fact, approaches using model 2 presented a greater overall accuracy on parameter estimation. In addition, the approach HRTO-HEKF-RELS with $\alpha = 0.99$ was very similar to the HRTO-HEKF-fixed for both models, which is a reflection of the low adaptability capacity of the RELS strategy with a high value of the forgetting factor. For the approaches HRTO-HEKF-RELS with $\alpha = 0.5$ and HRTO-HEKF-adaptive, the higher adaptability capacity was able to reduce the nMSE for the output variables, but presented a worse parameter estimation accuracy than the HRTO-HEKF-fixed, even though the models used in this approach were identified far from the operating point and in a fixed parameter scenario. It is noteworthy that the approach HRTO-HEKF-adaptive presented an undesirable oscillatory behavior, which appears due to the arise of oscillatory modes in the Hammerstein models during operation. This effect should be avoided and further investigated.

3.5 Conclusion

In this chapter, several different HRTO architectures are proposed based on the use of a Hammerstein model structure that combines the available static process model with a linear ARX identified from past data to provide approximate dynamics. Therefore, the requirement of availability of a rigorous dynamic process is removed from the original proposition of the HRTO, enabling it to be used in a large range of applications. The proposed methodologies showed satisfactory performances in parameter estimation and adequate computational costs. However, the proposed methodologies presented no resources to prevent the arise of

unstable and undesirable oscillatory modes on the Hammerstein model during the adaptive process, this should be further investigated.

Chapter 4

Steady-state real-time optimization using transient measurements in the absence of a dynamic mechanistic model: A framework of HRTO integrated with Adaptive Self-Optimizing IHMPC

A version of this chapter was published in the Journal of Process Control, Volume 106, October 2021, Pages 1-19 (DELOU *et al.*, 2021c).

4.1 Introduction

This chapter addresses one of the major backbones of the HRTO, which is the requirement of developing a dynamic process model introduced by the use of the dynamic observer. We overcome this issue by proposing an approximated dynamic model based on the available reliable static model. The assumption of an available static model is reasonable since its development is considered in every RTO design. We propose a Hammerstein structure that takes advantage of the static model with added dynamics by a linear autoregressive model (ARX) identified from plant data or experiments. It is important to emphasize that this identification procedure is very similar to the one used for obtaining classic internal models for MPCs (NAJIM and IKONEN, 2001) but with a different model structure. We propose and compare three models based on this structure, in which two present the interesting ability to match the steady states of the available static

model. It has been shown that Hammerstein models are able to present high fidelity to approximate dynamic models and it is suitable as an internal model for control purposes (RIBEIRO and SECCHI, 2019).

The Hammerstein model is introduced to present an EKF formulation to proceed with the parameters and measured output estimations, denoted Hammerstein EKF (HEKF). The parameters estimated using transient data are applied in the steady-state model adaptation, which concerns both HRTO and an adaptive Infinite Horizon MPC (IHMPC) that uses the proposed Hammerstein structure as its internal model. Additionally, the economic objectives are introduced in the control layer through the application of self-optimizing variables, instead of directly controlling the objective function as a virtual controlled variable.

Integrating MPC with SOC is also a manner to introduce the economic objective into the control layer, but this is done indirectly by controlling the self-optimizing variables. Therefore, this integration can be interpreted as an indirect EMPC. GRACIANO *et al.* (2015) proposed a complete framework that integrates SOC with a fixed-model MPC. The approach considered a static RTO integrated with a self-optimizing zone MPC, which tracks the setpoints of self-optimizing variables and keeps the active constraints within a zone. We expand this proposal by removing the classic RTO approach and introducing the HRTO based on the Hammerstein approximate dynamics using the proposed HEKF and the adaptive IHMPC as dynamic observers and controllers, respectively.

One of the main challenges of SOC is how to deal with changes in the set of active constraints since an active constraint is a natural self-optimizing variable. MANUM and SKOGESTAD (2012) proposed an approach based on parametric programming. Every active constraint region is determined *a priori*, and the process variables are monitored to determine the moment to switch between regions. Compared to MANUM and SKOGESTAD (2012), our approach presents the same advantages pointed out by GRACIANO *et al.* (2015): no need to solve large parametric programming problems and to store the results; no need to implement logic to determine when and how to switch between active constraints regions; no heuristics to handle nonlinearities; no requirement for extra measurements to smoothly change between active regions far from each other; and, no scheme to recover from wrong active set changes.

In addition, our approach also presents some advantages over the one proposed by GRACIANO *et al.* (2015):

- No wait time between optimization runs guarantees that the set of active constraints is always known;
- Naturally handling both types of active set changes: when a new constraint

becomes active and when an active constraint becomes inactive;

- The dynamic observer coupled with the adaptive IHMPC provides the system with more robustness to large drifting in parameters and unmeasured disturbances compared to a simple linear MPC.

In contrast with GRACIANO *et al.* (2015), we track the setpoints of all self-optimizing variables and active constraints. To do so, we propose a strategy to handle changes in the active set of constraints. Moreover, our approach presents full compatibility between the models used in the observer, controller, and optimization layers. Therefore, no model compatibility scheme is required.

In summary, the present chapter proposes a complete control framework by integrating the HRT0 methodology with a self-optimizing adaptive IHMPC. The main contributions of the framework are:

- Enabling the use of HRT0 in the absence of a reliable mechanistic dynamic model;
- Inclusion of the economic objectives into the controller layer through the self-optimizing variables, which allows a more reliable and robust control layer;
- There is full compatibility between the models used in the observer, controller, and optimizer;
- The framework naturally handles changes in the active constraint set, both the detection of a new constraint and the detection of the deactivation of an active constraint.

The remainder of the chapter is organized as follows. Section 4.2 presents the Hammerstein structure proposed as an approximate dynamic model to enable the application of HRT0 in the absence of a mechanistic dynamic process model. It also discusses three proposed models based on these structures, and the linearization and state-space formulation, which is the basis for the IHMPC. Section 4.3 presents the proposed integration of HRT0 and the self-optimizing adaptive IHMPC. It also analyses each step of the framework: the observer, the controller, and the optimizer formulations, and describes the SOC methodology and the strategy to handle changes in the active constraints set. Section 4.4 presents the Williams-Otto reactor case study and the results obtained. Finally, Section 4.5 presents the conclusions of the chapter. Moreover, Appendix A presents the formulation of the terminal weight of the self-optimizing IHMPC, Appendix B presents the implementation of the proposed controller as a QP and Appendix C is the Supplementary Material of this chapter.

4.2 Hammerstein structure as an approximate dynamic model

All past works regarding HRT0 assumed that both static and dynamic models of the process are available (KRISHNAMOORTHY *et al.*, 2018a; MATIAS *et al.*, 2018; SANTOS *et al.*, 2021; VALLURU and PATWARDHAN, 2019; VALLURU *et al.*, 2015). However, the assumption of the availability of a reliable dynamic mechanistic model might be unrealistic for most processes. In addition, the design of a dynamic model would significantly increase the project duration time, and costs, which could even make the application infeasible in practice.

It is assumed that an adequate steady-state model is available:

$$\mathbf{0} = \mathcal{F}(\bar{\mathbf{x}}^s, \bar{\mathbf{u}}, \bar{\mathbf{d}}) \quad (4.1a)$$

$$\bar{\mathbf{y}}^s = \mathcal{H}(\bar{\mathbf{x}}^s, \bar{\mathbf{u}}, \bar{\mathbf{d}}) \quad (4.1b)$$

in which, $\bar{\mathbf{x}}^s \in \mathbb{R}^{n_x}$ is the state vector at steady state, $\bar{\mathbf{y}}^s \in \mathbb{R}^{n_y}$ is the measured vector at steady state, $\bar{\mathbf{u}} \in \mathbb{R}^{n_u}$ is the input vector and $\bar{\mathbf{d}} \in \mathbb{R}^{n_d}$ is the parameters and unmeasured disturbances vector in which uncertainty occurs. Herein, the overlying bar ($\bar{\cdot}$) represents positional variables, in contrast with variables deviated from a reference point, which will be denoted without this emphasis. In addition, $\mathcal{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_x}$ and $\mathcal{H} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_y}$ are the static mapping of the states and the outputs related to the inputs and parameters, respectively. For the sake of simplifying notation without loss of generality, this model will be represented as:

$$\bar{\mathbf{y}}^s = \bar{\mathbf{y}}^s(\bar{\mathbf{u}}, \bar{\mathbf{d}}) \quad (4.2)$$

A general Hammerstein structure is defined by a nonlinear processing of an input signal followed by a linear processing in order to provide an output signal. Here, this structure is exploited so that the nonlinear processing consists of the available static model and the linear processing consists of a linear ARX with feedback to introduce a pseudo-dynamics. The general proposed structure can be visualized in its block diagram form in Figure 4.1 and, given a nominal point $(\bar{\mathbf{y}}, \bar{\mathbf{u}}, \bar{\mathbf{d}})_{nom}$, its recursion form can be written as:

$$\mathbf{y}_{k+1}^h = \mathbf{A}^h \mathbf{y}_k^h + \mathbf{B}^h \mathbf{y}_k^s(\bar{\mathbf{u}}_k, \bar{\mathbf{d}}_k) \quad (4.3)$$

in which, $\mathbf{y}_{k+1}^h := \bar{\mathbf{y}}_{k+1}^h - \bar{\mathbf{y}}_{nom} \in \mathbb{R}^{n_y}$ is the set of measured variables predicted

by the Hammerstein structure at an instant k based on the past prediction $\mathbf{y}_k^h := \bar{\mathbf{y}}_k^h - \bar{\mathbf{y}}_{nom}$ and the actual modeled static value $\mathbf{y}_k^s(\bar{\mathbf{u}}_k, \bar{\mathbf{d}}_k) := \bar{\mathbf{y}}_k^s(\bar{\mathbf{u}}_k, \bar{\mathbf{d}}_k) - \bar{\mathbf{y}}_{nom}$.

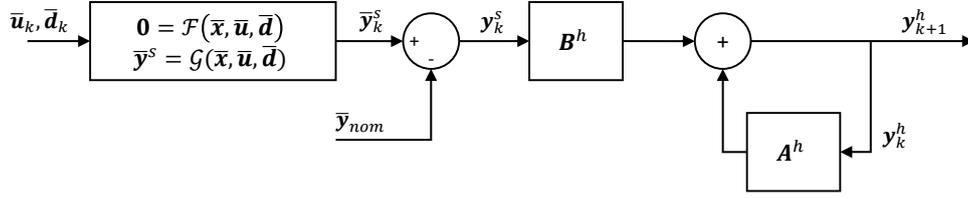


Figure 4.1: Block diagram of the proposed Hammerstein structure.

Matrices \mathbf{A}^h and \mathbf{B}^h have dimension $n_y \times n_y$ and are called dynamic matrices. They should be identified from past data or from identification experiments imposed to the process. Here, we propose three models based on the structure of the dynamic matrices, they are:

- model 1: $\mathbf{B}^h = \mathbf{I}_{n_y} - \mathbf{A}^h$, in which \mathbf{A}^h is a diagonal matrix, containing only time constants to be estimated;
- model 2: $\mathbf{B}^h = \mathbf{I}_{n_y} - \mathbf{A}^h$, in which \mathbf{A}^h is a full matrix, accounting for coupled dynamics between the measured variables;
- model 3: \mathbf{A}^h and \mathbf{B}^h are full matrices and independent of each other.

It is important to note that models 1 and 2 have the interesting property that their steady states match with the steady states of the static model, a property that is not observed in model 3. However, model 3 has the potential to provide better local approximations since it presents $2n_y^2$ parameters to describe the dynamics, while models 1 and 2 present n_y and n_y^2 parameters, respectively. Although this fact seems to be advantageous, the increasing number of parameters can hinder the identification procedure, demanding higher excitation in the identification experiments. In addition, despite presenting a better local approximation, model 3 might present significant steady-state errors concerning the static model when the operational conditions are far from the identification scenario.

In regards to the identification stage, several strategies are possible to obtain the dynamic matrices (NAJIM and IKONEN, 2001). The identification procedures are similar to what is done in classic linear MPC strategies to obtain the controller's internal linear models. The proposed methodology presents a similar prerequisite to classic MPC since the Hammerstein dynamic matrices must be previously identified to enable its online use in the observer, controller, and optimizer approaches. The identification stage goal is to minimize the distance between plant data and the model prediction. The main difference is the structure of the predictive model that, different from classic MPC internal models, takes advantage of

the available nonlinear static mapping of the process. Independently of the identification problem formulation, it is important to guarantee the model stability by the inclusion of the following nonlinear constraint in the identification problem:

$$\|\boldsymbol{\lambda}(\mathbf{A}^h)\| < 1 \quad (4.4)$$

in which, $\boldsymbol{\lambda}(\mathbf{A}^h)$ is the spectrum of matrix \mathbf{A}^h . Also, it might be the case to avoid the rise of oscillatory modes if the process does not present oscillatory behavior. In this case, the eigenvalues of matrix \mathbf{A}^h should not present imaginary terms, which can be guaranteed by the inclusion of the following constraint to the identification problem:

$$\text{Im}(\boldsymbol{\lambda}(\mathbf{A}^h)) = 0 \quad (4.5)$$

In the present study, the dynamic matrices are identified offline and considered fixed during real-time operation. Also, the inclusion of the constraints presented in Equations 4.4 and 4.5 may cause a non-convexity in the identification optimization problem, which burdens the solution-finding effort, one may consider using approaches such as global optimization solvers, or multi-start approaches for instance.

4.2.1 Linearization of the Hammerstein structure and State-Space formulation

The proposed Hammerstein structure, Equation 4.3, can be linearized with respect to a nominal point $(\bar{\mathbf{y}}, \bar{\mathbf{u}}, \bar{\mathbf{d}})_{nom}$. First, the nonlinear static model is linearized:

$$\mathbf{y}_k^s(\bar{\mathbf{u}}_k, \bar{\mathbf{d}}_k) \approx \mathbf{G}^u \mathbf{u}_k + \mathbf{G}^d \mathbf{d}_k \quad (4.6)$$

in which, $\mathbf{u}_k := \bar{\mathbf{u}}_k - \bar{\mathbf{u}}_{nom}$, $\mathbf{d}_k := \bar{\mathbf{d}}_k - \bar{\mathbf{d}}_{nom}$ and the matrices \mathbf{G}^u and \mathbf{G}^d are the gain matrices with respect to the inputs and the disturbances, respectively. The gain matrices can be calculated by:

$$\mathbf{G}^u := \left. \frac{\partial \mathbf{y}^s}{\partial \mathbf{u}} \right|_{nom} = \left[\frac{\partial \mathcal{H}}{\partial \mathbf{x}} \left(\frac{\partial \mathcal{F}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathcal{F}}{\partial \mathbf{u}} \right]_{nom} \quad (4.7a)$$

$$\mathbf{G}^d := \left. \frac{\partial \mathbf{y}^s}{\partial \mathbf{d}} \right|_{nom} = \left[\frac{\partial \mathcal{H}}{\partial \mathbf{x}} \left(\frac{\partial \mathcal{F}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathcal{F}}{\partial \mathbf{d}} \right]_{nom} \quad (4.7b)$$

Hence, the linear Hammerstein model can be written as:

$$\mathbf{y}_{k+1}^h = \mathbf{A}^h \mathbf{y}_k^h + \mathbf{B}^h \mathbf{G}^u \mathbf{u}_k + \mathbf{B}^h \mathbf{G}^d \mathbf{d}_k \quad (4.8)$$

Which is already in a positional state-space formulation. To introduce an incremental state-space, the equation $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$ is added in an augmented formulation:

$$\begin{bmatrix} \mathbf{y}_{k+1}^h \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A}^h & \mathbf{B}^h \mathbf{G}^u \\ \mathbf{0} & \mathbf{I}_{n_u} \end{bmatrix} \begin{bmatrix} \mathbf{y}_k^h \\ \mathbf{u}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{B}^h \mathbf{G}^u \\ \mathbf{I}_{n_u} \end{bmatrix} \Delta \mathbf{u}_k + \mathbf{B}^h \mathbf{G}^d \mathbf{d}_k \quad (4.9a)$$

$$\mathbf{y}_{k+1}^h = \begin{bmatrix} \mathbf{I}_{n_y} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{k+1}^h \\ \mathbf{u}_k \end{bmatrix} \quad (4.9b)$$

Introducing new notation, Equation 4.9 can be rewritten as:

$$\mathbf{y}_{k+1} = \mathbf{A} \mathbf{y}_k + \mathbf{B}^u \Delta \mathbf{u}_k + \mathbf{B}^d \mathbf{d}_k \quad (4.10a)$$

$$\mathbf{y}_{k+1}^h = \mathbf{C} \mathbf{y}_{k+1} \quad (4.10b)$$

It is important to emphasize, for clarity of notation, that matrix \mathbf{A} represents the augmented transition state in the incremental linear state space and the Hammerstein dynamic matrices, \mathbf{A}^h and \mathbf{B}^h , as previously mentioned, are fixed. Therefore, any updating index associated with the matrix, such as \mathbf{A}_k , represents an update in the linearization matrices, \mathbf{G}^u and \mathbf{G}^d .

These results will be used in the formulation of the parameter estimation and control strategy, described in Sections 4.3.1 and 4.3.2, respectively. The proposed Hammerstein structure can be used as an alternative for the case where a dynamic mechanistic model is unavailable and for any strategy that relies on the linear state-space formulation. As shown in Equations 4.8 and 4.10, any observer, controller, or optimization strategy that is based on directional or incremental linear state-space models could, in principle, be developed for the proposed Hammerstein model. For example, the Extended Kalman Filter and Infinite Horizon MPC developed in the present chapter.

4.3 Real-time Adaptive Self-Optimizing Control Framework

A scheme of the proposed HRTO framework can be visualized in Figure 4.2. The whole framework is based on the assumption that a mechanistic dynamic model is not available, but a reliable static model of the plant and the dynamic matrix

ces of the Hammerstein model, which need to be previously identified from plant data or plant experiments, are at hand. It is used in the proposed Hammerstein structure to obtain an approximate dynamic model identified around a single operating point.

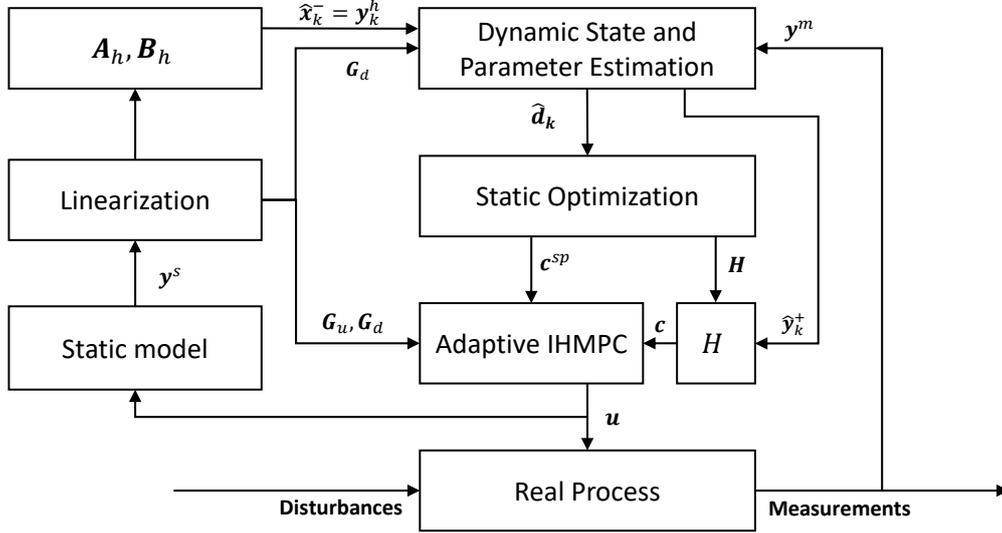


Figure 4.2: Proposed Real-time Adaptive Self-Optimizing Control Framework.

Each stage of the proposed framework is further detailed in the following subsections. This section is organized as Subsection 4.3.1 details the complete description of the dynamic state and parameter estimation based on the use of the Hammerstein model inside an EKF formulation; Subsection 4.3.2 presents the complete formulation of the proposed Self-Optimizing Adaptive IHMPC, including the formulation of the control problem; Subsection 4.3.3 describes the economic optimization problem considered in the Static Optimization layer; and, finally, Subsection 4.3.4 details the approach used to calculate the H-matrix of the SOC strategy and the proposed scheme to handle changes in the set of active constraints.

The HRTO cycle starts by sampling measurements, regardless of their transient or steady nature, and using them in a dynamic state and parameter estimator. These estimations, which include uncertain model parameters and unmeasured disturbances, are used to adapt the static model of the economic optimization layer and the linearized Hammerstein model of the controller and parameter estimator.

The static economic optimization is responsible for obtaining the setpoints of the self-optimizing variable c^{sp} and determining the active set of constraints, which will be used to construct the combination matrix H . In the controller layer, an adaptive IHMPC is responsible for obtaining the optimal input trajectory that minimizes the quadratic distance between the self-optimizing variables and their economic optimum setpoints while keeping the active constraints at their optimal bounds. The prediction of the controller is performed with a single model,

based on the adapted linearized Hammerstein structure, and the optimal input trajectory is implemented in a receding horizon manner. Unlike the classic RTO scheme, in which the optimization layer runs in a different time scale from the control layer due to the SSD stage, in both HRTO and the proposed framework illustrated in Figure 4.2, all the blocks run in the same time scale. In other words, all the blocks are synchronized and run in every time instant k .

4.3.1 Parameter Estimation based on transient measurements

The dynamic parameter estimator is the main component that differentiates the promising HRTO methodology from classic RTO. KRISHNAMOORTHY *et al.* (2018a) proposed the use of an EKF and pointed out that, in principle, any strategy would be suitable for this purpose. VALLURU and PATWARDHAN (2019) argued that both recursive and optimization-based approaches would be suitable for the task. Examples of recursive approaches are the EKF, the Unscented Kalman Filter (UKF) (SIMON, 2006b), and the Particle Filter (PF) (JOUIN *et al.*, 2016). Among the optimization-based methods, the Moving Horizon Estimation (MHE) (PATWARDHAN *et al.*, 2012) is the most prominent method. However, one must consider the trade-off between the estimation quality and computational cost of each approach. In this matter, a well-tuned EKF is the most suitable approach since it provides a fair estimation accuracy with a significantly low computational effort, even when compared to other conventional approaches, such as the Constrained EKF and the UKF. This result is observed in several applications (ALEXANDER *et al.*, 2020; APIO *et al.*, 2019; HASELTINE and RAWLINGS, 2005; KALLENBERGER *et al.*, 2007; LAVIOLA, 2003; ST-PIERRE and GINGRAS, 2004; STELZER *et al.*, 2017). It is also worth mentioning the EKF with Rauch-Tugh-Striebel (RTS) smoother proposed by SALAU *et al.* (2012), which presents characteristics of an MHE but with the EKF computational effort.

Here, we propose an EKF formulation based on the Hammerstein model, which is referred to as Hammerstein EKF (HEKF). To accomplish the parameter and state estimation simultaneously, an augmented state variable is proposed where the instant dynamic hypothesis is assigned to the parameters. Hence, the *a priori* prediction equations are:

$$\hat{\mathbf{x}}_k^- := \begin{bmatrix} \hat{\mathbf{y}}_k^- \\ \hat{\mathbf{d}}_k^- \end{bmatrix} = \begin{bmatrix} \mathbf{A}^h \hat{\mathbf{y}}_{k-1}^+ + \mathbf{B}^h \mathbf{y}^s(\bar{\mathbf{u}}_{k-1}, \hat{\mathbf{d}}_{k-1}^+) + \mathbf{w}_{y,k} \\ \hat{\mathbf{d}}_{k-1}^+ + \mathbf{w}_{d,k} \end{bmatrix} \quad (4.11)$$

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1}^+ \mathbf{F}_k^T + \mathbf{Q} \quad (4.12)$$

in which, $\mathbf{w}_{y,k}$ and $\mathbf{w}_{d,k}$ are artificial zero-mean noise, so that $\mathbf{w}_{y,k} \sim \mathcal{N}(0, \mathbf{Q}_y)$ and $\mathbf{w}_{d,k} \sim \mathcal{N}(0, \mathbf{Q}_d)$; the hat emphasis ($\hat{\cdot}$) denotes estimated quantities and the minus and plus sign superscripts, $(\cdot)^-$ and $(\cdot)^+$, denote *a priori* and *a posteriori* estimated quantities; matrix $\mathbf{Q} := \text{diag}(\mathbf{Q}_y, \mathbf{Q}_d)$ is the augmented noise covariance matrix, \mathbf{P}_k is the augmented covariance estimate matrix and \mathbf{F}_k is the augmented state transition matrix defined as:

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{A}^h & \mathbf{B}^h \mathbf{G}_k^d \\ \mathbf{0}_{n_d \times n_y} & \mathbf{I}_{n_d \times n_d} \end{bmatrix} \quad (4.13)$$

in which, \mathbf{G}_k^d is reevaluated every instant following Equation 4.7b, by shifting the nominal point as $(\bar{\mathbf{y}}, \bar{\mathbf{u}}, \bar{\mathbf{d}})_{nom} := (\bar{\mathbf{y}}^s(\bar{\mathbf{u}}, \hat{\mathbf{d}}^+), \bar{\mathbf{u}}, \hat{\mathbf{d}}^+)_{k-1}$. Finally, the *a posteriori* correction equations, based on the measurement update, are:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{\Gamma}^T (\mathbf{\Gamma} \mathbf{P}_k^- \mathbf{\Gamma}^T + \mathbf{R})^{-1} \quad (4.14)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k^m - \hat{\mathbf{y}}_k^-) \quad (4.15)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{\Gamma}) \mathbf{P}_k^- \quad (4.16)$$

$$\hat{\mathbf{y}}_k^+ = \mathbf{\Gamma} \hat{\mathbf{x}}_k^+ \quad (4.17)$$

in which, \mathbf{K}_k is the Kalman gain matrix, \mathbf{R} is the measurement covariance matrix and $\mathbf{\Gamma}$ is the observation matrix defined as:

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{I}_{n_y \times n_y} & \mathbf{0}_{n_y \times n_d} \end{bmatrix} \quad (4.18)$$

An interesting characteristic of the EKF methodology is that matrices \mathbf{Q} and \mathbf{R} can be considered as filter tuning parameters to balance the trade-off between model prediction and measurement update. By choosing $\mathbf{Q} \prec \mathbf{R}$, the filter prioritizes model prediction over measurement update, and filter estimates become less noisy. On the contrary, $\mathbf{R} \prec \mathbf{Q}$, the filter estimates prioritize the measurements and become noisier (SIMON, 2006b).

4.3.2 Adaptive Self-Optimizing IHMPC based on the Hammerstein structure

The proposed adaptive linear MPC uses the linear Hammerstein model as its internal model and explicitly tracks the self-optimizing variables in an incremental input form. The adaptive infinite-horizon control problem is defined by Equation

4.19 subjected to the set of linear constraints.

$$\begin{aligned}
\min_{\substack{\Delta \mathbf{u}, \boldsymbol{\rho}_y^L, \boldsymbol{\rho}_y^U, \\ \boldsymbol{\rho}_u^L, \boldsymbol{\rho}_u^U}} J_k = & \sum_{j=1}^N (\mathbf{c}_{k+j} - \mathbf{c}_k^{sp})^T \mathbf{W}_c (\mathbf{c}_{k+j} - \mathbf{c}_k^{sp}) + \mathbf{w}_{\rho_y^L}^T \boldsymbol{\rho}_{y,j}^L + \mathbf{w}_{\rho_y^U}^T \boldsymbol{\rho}_{y,j}^U + \\
& \sum_{j=0}^{N-1} \Delta \mathbf{u}_{k+j}^T \mathbf{W}_{\Delta u} \Delta \mathbf{u}_{k+j} + \mathbf{w}_{\rho_u^L}^T \boldsymbol{\rho}_{u,j}^L + \mathbf{w}_{\rho_u^U}^T \boldsymbol{\rho}_{u,j}^U + \\
& \mathbf{c}_{k+N}^T \mathbf{P} \mathbf{c}_{k+N}
\end{aligned} \tag{4.19a}$$

$$\text{s.t. } \forall j \in [1, 2, \dots, N-1]$$

$$\mathbf{y}_{k+j}^h = \mathbf{C} \mathbf{A}_k \mathbf{y}_{k+j-1} + \mathbf{C} \mathbf{B}_k^u \Delta \mathbf{u}_{k+j-1} + \mathbf{C} \mathbf{B}_k^d \mathbf{d}_{k+j-1} \tag{4.19b}$$

$$\mathbf{y}_k^h = \hat{\mathbf{y}}_k^+ \tag{4.19c}$$

$$\mathbf{c}_{k+j} = \mathbf{H} \mathbf{y}_{k+j}^h \tag{4.19d}$$

$$\Delta \mathbf{u}^L \leq \Delta \mathbf{u}_{k+j} \leq \Delta \mathbf{u}^U \tag{4.19e}$$

$$\mathbf{y}^L - \boldsymbol{\rho}_{y,j}^L \leq \mathbf{y}_{k+j}^h \leq \mathbf{y}^U + \boldsymbol{\rho}_{y,j}^U \tag{4.19f}$$

$$\mathbf{u}^L - \boldsymbol{\rho}_{u,j}^L \leq \mathbf{u}_{k+j+1} \leq \mathbf{u}^U + \boldsymbol{\rho}_{u,j}^U \tag{4.19g}$$

$$\boldsymbol{\rho}_{y,j}^L \geq \mathbf{0}, \boldsymbol{\rho}_{y,j}^U \geq \mathbf{0}, \boldsymbol{\rho}_{u,j}^L \geq \mathbf{0}, \boldsymbol{\rho}_{u,j}^U \geq \mathbf{0} \tag{4.19h}$$

in which, m is the control horizon that defines the discretization of the control actions, defined as $\Delta \mathbf{u} := [\Delta \mathbf{u}_k, \dots, \Delta \mathbf{u}_{k+N-1}]$; J_k is the objective function at the k^{th} run of the control problem; \mathbf{W}_c and $\mathbf{W}_{\Delta u}$ are diagonal semi-positive weight matrices for the self-optimizing variables and the variation of the inputs, respectively.

Equation 4.19b is the linearized Hammerstein adaptive predictive model derived in Section 4.2.1; the model matrices \mathbf{A}_k , \mathbf{B}_k^u and \mathbf{B}_k^d are updated outside the control problem by shifting the nominal point of the linearization, as it is done in the HEKF and discussed in Section 4.3.1. A distinguishing feature of the proposed controller that differs from classic linear MPCs is its adaptability capacity introduced by the successive linearization of the static process mapping used in the Hammerstein model. This formulation allows for a control performance similar to an NMPC with the low computational effort of a classic linear MPC (DI MARCO *et al.*, 1997).

The initial condition of the self-optimizing variables in the control problem given in Equation 4.19 is the *a posteriori* estimation of the measured variables, as denoted in Equation 4.19c. Equations 4.19e-4.19g represent, respectively, the hard constraints on the input variations, the soft constraints on the measured variables, and the soft constraints on the input variables. In which, the superscripts $(\cdot)^L$ and $(\cdot)^U$ denote the lower and upper bounds, respectively. The constraints are

softened by the addition of slack variables as degrees of freedom in the control problem in order to enhance numerical robustness and prevent infeasible solutions. The added slack variables are defined as $\boldsymbol{\rho}_y^L := [\boldsymbol{\rho}_{y,1}^L, \dots, \boldsymbol{\rho}_{y,N}^L] \in \mathbb{R}^{Nn_y}$, $\boldsymbol{\rho}_y^U := [\boldsymbol{\rho}_{y,1}^U, \dots, \boldsymbol{\rho}_{y,N}^U] \in \mathbb{R}^{Nn_y}$, $\boldsymbol{\rho}_u^L := [\boldsymbol{\rho}_{u,1}^L, \dots, \boldsymbol{\rho}_{u,N}^L] \in \mathbb{R}^{Nn_u}$ and $\boldsymbol{\rho}_u^U := [\boldsymbol{\rho}_{u,1}^U, \dots, \boldsymbol{\rho}_{u,N}^U] \in \mathbb{R}^{Nn_u}$. ℓ_1 penalties are added to the objective function of the control problem by the weights vectors $\mathbf{w}_{\rho_y^L}$, $\mathbf{w}_{\rho_y^U}$, $\mathbf{w}_{\rho_u^L}$ and $\mathbf{w}_{\rho_u^U}$. It is safe to say that, if there is a feasible solution for the hard problem, it will match the soft problem solution for a sufficiently large value of the ℓ_1 penalties weights. Here, the slack variables are added in both input and measured variables for the sake of generality. However, it is often useful to implement hard constraints to some input variables, especially when they represent physical bounds such as actuator saturation. It can be done without changing the control problem presented in Equation 4.19 by setting the slacks' weights to zero and making their upper and lower bounds equal to zero. This way, hard constraints can be implemented even in the presence of slack variables in the control problem. However, if an input variable reaches one of its bounds, this could generate undesirable offsets between the actual operation from its optimum setpoints due to the loss of controllability caused by an input saturation.

In addition, the matrix \mathbf{P} is the terminal cost that is calculated by the following discrete Lyapunov equation:

$$\mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{P} + (\mathbf{H}\mathbf{C})^T \mathbf{W}_c \mathbf{H}\mathbf{C} = \mathbf{0} \quad (4.20)$$

The proof of Equation 4.20 can be found in Appendix A. To the best of the authors' knowledge, this formulation is unprecedented as no other study that explicitly handles the self-optimizing variables in an infinite horizon formulation was found.

Problem 4.19 is implemented as a Quadratic Program (QP), this formulation can be found in Appendix B.

4.3.3 Steady-state Economic Optimization

Provided the availability of a reliable static model, described in Equation 4.1, an optimization problem aiming to minimize a cost function, or maximize a profit function, can be formulated as in Problem 4.21.

$$\bar{\mathbf{u}}_{k+1}^* = \arg \min_{\bar{\mathbf{u}}, \bar{\mathbf{x}}^s} J_{eco}(\bar{\mathbf{y}}^s, \bar{\mathbf{u}}, \hat{\mathbf{d}}_k^+) \quad (4.21a)$$

$$\text{s.t. } \mathbf{0} = \mathcal{F}(\bar{\mathbf{x}}^s, \bar{\mathbf{u}}, \hat{\mathbf{d}}_k^+) \quad (4.21b)$$

$$\bar{\mathbf{y}}^s = \mathcal{H}(\bar{\mathbf{x}}^s, \bar{\mathbf{u}}, \hat{\mathbf{d}}_k^+) \quad (4.21c)$$

$$\bar{\mathbf{y}}^L \leq \bar{\mathbf{y}}^s \leq \bar{\mathbf{y}}^U \quad (4.21d)$$

$$\bar{\mathbf{u}}^L \leq \bar{\mathbf{u}} \leq \bar{\mathbf{u}}^U \quad (4.21e)$$

in which, function $J_{eco} : \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}$ is a nonlinear economic function to be minimized. Both inputs and steady-state variables are degrees of freedom in this formulation, so the model enters as the equality constraints in Equations 4.21b and 4.21c. In addition, the bounds of measured and input variables, Equations 4.21d and 4.21e, are compatible with the bounds of the MPC problem. The model is directly adapted by the set of uncertain parameters estimated by the dynamic estimator, $\hat{\mathbf{d}}_k^+$.

Here, we consider a reliable model to match the plant optimality conditions once the set of uncertain parameters is correctly estimated. In other words, a model that meets the model adequacy requirements (FORBES and MARLIN, 1994, 1996; FORBES *et al.*, 1994).

The economic optimization problem runs synchronously with the observer and the controller. Once an optimal solution is obtained for the next instant, $\bar{\mathbf{u}}_{k+1}^*$, it is used to obtain the optimal setpoints of the original output variables:

$$\bar{\mathbf{y}}_{k+1}^{sp} = \bar{\mathbf{y}}^s(\bar{\mathbf{u}}_{k+1}^*, \hat{\mathbf{d}}_k^+) \quad (4.22)$$

These setpoints are then used to obtain the setpoints of the current self-optimizing variables. It is interesting to note that, in the proposed methodology, there will be total compatibility between the models used in the observer, controller, and optimization layers if one chooses to use the Hammerstein models 1 or 2. That is important to notice, especially for the setpoint compatibility between the optimization and controller layer since the setpoints obtained in the optimization will always be feasible for the internal linear models of the MPC layer. This fact removes the need for any intermediate compatibilization problem, such as it is done in LP-MPC, QP-MPC, or any heuristics to prevent infeasible solutions in the MPC due to poor predictive accuracy of the linear models (ROTAVA and ZANIN, 2005; SORENSEN and CUTLER, 1998; YING and JOSEPH, 1999).

4.3.4 Obtaining Self-Optimizing variables and handling changes in Active Constraints

Among the available methods for obtaining the self-optimizing variables, the local methods present greater potential for real-time applications over the earlier

brute force methods (SKOGESTAD, 2000). JÄSCHKE *et al.* (2017) made a thorough review of the progress of Self-Optimizing Control. The most prominent methods are the Null-space method (ALSTAD and SKOGESTAD, 2007); the Extended Null-space method (ALSTAD *et al.*, 2009); and the Minimum Loss method (YELCHURU and SKOGESTAD, 2012).

In this chapter, we consider the Minimum Loss method to determine the combination matrix H . This method presents some advantages, such as providing an optimal trade-off between measurement noise and rejecting the disturbances. It can find the best local combination of measurements that minimizes the loss function and has an explicit solution. That said, the H-matrix can be determined by:

$$H = (G^u)^T (Y Y^T)^{-1} \quad (4.23)$$

Introducing the scaling matrices W_d and W_{w_y} , that contains the magnitude of the disturbances and the measurement noise, matrix Y is defined as:

$$Y := \begin{bmatrix} F W_d & W_{w_y} \end{bmatrix} \quad (4.24)$$

in which, F is the sensitivity matrix of the optimal measurements in relation to the disturbances. It can be calculated by finite differences (GRACIANO *et al.*, 2015), automatic differentiation, nonlinear programming sensitivity (PIRNAY *et al.*, 2012) or, as done in the present chapter, by the local approximation:

$$F = G^d - G^u J_{uu}^{-1} J_{ud} \quad (4.25)$$

in which J_{uu} and J_{ud} are the Hessians of the optimization cost function evaluated at the nominal point.

However, the above methodology is only valid for the variables which do not present an active constraint. Since if there are variables in one of their bounds in the optimal solution of Problem 4.21, these variables are naturally the self-optimizing variable. Due to this reason, handling active constraint set changes in real-time operation has been a topic of interest in the field of SOC (KRISHNAMOORTHY and SKOGESTAD, 2020; MANUM and SKOGESTAD, 2012). Again, we refer to JÄSCHKE *et al.* (2017) who dedicated a full section about this topic.

In the present methodology, changes in the active constraint set are naturally handled. The optimization problem runs synchronously with the control problem, which is enabled due to the efficient dynamic observer approach. In contrast

with the classic RTO strategy, in which there are long wait periods between each run of the optimizer due to steady-state detection; here, the active set related to the estimated parameters is instantly known and can be used to determine a new combination matrix \mathbf{H} every time instant.

Let \mathbb{Y}_k^a be the set of all n_y^a active constraints, which is evaluated at every time instant k based on the previous run of the economic optimization problem. If this set is found to be empty, the H-matrix is simply evaluated based on Equations 4.25, 4.24, and 4.23. Otherwise, the construction of the H-matrix must be reconfigured to remove n_y^a self-optimizing variables from the approach since the active constraints are natural self-optimizing variables.

In the SOC methodology, the number of self-optimizing variables is given by the number of input variables. Therefore, to reduce the number of self-optimizing variables in case n_y^a output variables are active, n_y^a input variables must be removed from the procedure to obtain \mathbf{H} . For doing so, there has to be a criterion to choose which input variable will be removed for each active constraint. In the present chapter, we remove the input variables with the highest gain associated with each active constraint based on matrix \mathbf{G}_k^u evaluated at each time instant k . Thus, the matrices \mathbf{G}_k^u , \mathbf{G}_k^d , $\mathbf{J}_{uu,k}$, $\mathbf{J}_{ud,k}$ and $\mathbf{W}_{wy,k}$ are reconfigured in order to remove the columns and rows associated to the active variables and the removed input variables, respectively. Afterward, the H-matrix is also evaluated based on Equations 4.25, 4.24, and 4.23, but with the reconfigured matrices. The resulting matrix dimension is $(n_u - n_y^a) \times (n_y - n_y^a)$, coping only with the linear combination of all output variables that are not active. Therefore, this matrix should be further tailored to present dimension $n_u \times n_y$, and this is done by adding the null columns associated with the active output variables and null rows related to the removed inputs, the only inserted unitary element must be related to the active variable and its associated input. This way, part of the resulting H-matrix is responsible for the linear combination of the nonactive variables, and part is responsible for selecting the active variables. This procedure is illustrated in an algorithmic fashion in Algorithm 1.

In Algorithm 1, for the sake of simplifying notation, we defined two matrix operations to represent the removal of rows and columns and the inclusion of null rows and columns. $\mathbf{M}^{-(i,j)}$, $\mathbf{M}^{-(i,\cdot)}$ and $\mathbf{M}^{-(\cdot,j)}$ represent the removal of row i and column j from matrix \mathbf{M} , the removal of row i from matrix \mathbf{M} and the removal of column j from matrix \mathbf{M} , respectively. The notation $\mathbf{M}^{+(i,j)}$ represents the inclusion of a null row and a null column in the resulting positions i and j . Therefore, if \mathbf{M} has dimensions $n \times m$, the resulting matrix $\mathbf{M}^{+(i,j)}$ has dimensions $(n+1) \times (m+1)$, in which row i and column j have only null elements. In addition, $M(i, j)$ represents the element in row i and column j of the matrix \mathbf{M} .

Algorithm 1: Reconfiguration of \mathbf{H} under changes in Active Constraint at time instant k

Evaluate the set of active constraint \mathbb{Y}_k^a based on $(\bar{\mathbf{u}}_k^*, \bar{\mathbf{y}}_k^*)$;
 $\mathbf{J}_{uu,k}, \mathbf{J}_{ud,k} \leftarrow$ Evaluate Hessian matrices based on $(\bar{\mathbf{u}}_k^*, \bar{\mathbf{y}}_k^*)$;
if $\mathbb{Y}_k^a = \emptyset$ **then**
 $\mathbf{F}_k \leftarrow \mathbf{G}_k^d - \mathbf{G}_k^u \mathbf{J}_{uu,k}^{-1} \mathbf{J}_{ud,k}$;
 $\mathbf{Y}_k \leftarrow [\mathbf{F}_k \mathbf{W}_d \quad \mathbf{W}_{wy}]$;
 $\mathbf{H}_k \leftarrow (\mathbf{G}_k^u)^T (\mathbf{Y}_k \mathbf{Y}_k^T)^{-1}$;
else
 $\mathbf{G}_k^{u,a} \leftarrow \mathbf{G}_k^u$; $\mathbf{G}_k^{d,a} \leftarrow \mathbf{G}_k^d$;
 $\mathbf{J}_{uu,k}^a \leftarrow \mathbf{J}_{uu,k}$; $\mathbf{J}_{ud,k}^a \leftarrow \mathbf{J}_{ud,k}$;
 $\mathbf{W}_{wy,k}^a \leftarrow \mathbf{W}_{wy}$;
 for each $y^a \in \mathbb{Y}_k^a$ **do**
 $i \leftarrow \{q : y^a = y_q, \forall q \in 1, \dots, n_y\}$;
 $j \leftarrow \{q : \arg \max_q |G_k^u(i, q)|, \forall q \in 1, \dots, n_u\}$;
 $\mathbf{G}_k^{u,a} \leftarrow \mathbf{G}_k^{u,a-(i,j)}$; $\mathbf{G}_k^{d,a} \leftarrow \mathbf{G}_k^{d,a-(i,\cdot)}$;
 $\mathbf{J}_{uu,k}^a \leftarrow \mathbf{J}_{uu,k}^{a-(j,j)}$; $\mathbf{J}_{ud,k}^a \leftarrow \mathbf{J}_{ud,k}^{a-(\cdot,j)}$;
 $\mathbf{W}_{wy,k}^a \leftarrow \mathbf{W}_{wy,k}^{a-(i,i)}$;
 end
 $\mathbf{F}_k^a \leftarrow \mathbf{G}_k^{d,a} - \mathbf{G}_k^{u,a} (\mathbf{J}_{uu,k}^a)^{-1} \mathbf{J}_{ud,k}^a$;
 $\mathbf{Y}_k^a \leftarrow [\mathbf{F}_k^a \mathbf{W}_d \quad \mathbf{W}_{wy,k}^a]$;
 $\mathbf{H}_k \leftarrow (\mathbf{G}_k^{u,a})^T (\mathbf{Y}_k^a (\mathbf{Y}_k^a)^T)^{-1}$;
 for each $y^a \in \mathbb{Y}_k^a$ **do**
 $i \leftarrow \{q : y^a = y_q, \forall q \in 1, \dots, n_y\}$;
 $j \leftarrow \{q : \arg \max_q |G_k^u(i, q)|, \forall q \in 1, \dots, n_u\}$;
 $\mathbf{H}_k \leftarrow \mathbf{H}_k^{+(j,i)}$;
 $H_k(j, i) \leftarrow 1$;
 end
end

The proposed framework presents some interesting and distinguished features, so the main contributions of the methodology are:

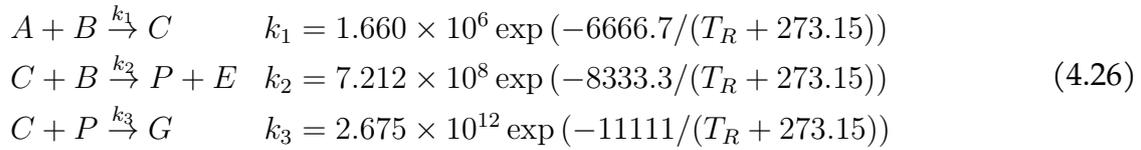
- Enables the application of the HRT0 methodology in the absence of a reliable dynamic mechanistic process model;
- The online estimator can filter noisy measurements and estimate model parameters and unmeasured disturbances, which are used in both steady-state optimization and adaptive IHMPC, regardless of the stationarity of the measurements;

- The optimization framework can achieve the minimum possible economic loss even in the scenario of inaccurate disturbance estimation due to the control of the self-optimizing variables;
- Active constraint changes are handled by frequent optimization runs and by the setpoint tracking strategy in the control layer.

4.4 Case Study: The Williams-Otto Reactor

The Williams-Otto reactor is a classic benchmark for real-time optimization studies. It was first proposed by WILLIAMS and OTTO (1960) and later modified by ROBERTS (1979). The case study consists of a CSTR, and its versatility comes from the definition of two sets of reactions, one containing 3 reactions and the other containing 2 reactions. They are usually used as plant and model to simulate structural plant-model mismatch (MARCHETTI *et al.*, 2016).

In the present study, as we are focusing primarily on parametric uncertainty, structural uncertainty will be tackled in future work. The 3 reactions model is considered as plant and model. Their reactions and kinetic equations are:



in which components A and B are fed to the reactor with mass flow rates F_A and F_B , respectively. E and P are the desired products, G is an undesired product, and C is an intermediate component. It is assumed that the reactor has perfect temperature control, with temperature T_R , and it keeps its mass constant, so the output flow rate is $F = F_A + F_B$, with mass holdup W .

In this chapter, it is assumed that only the static model is available:

$$0 = F_A + Fw_A - Wk_1w_Aw_B \tag{4.27a}$$

$$0 = F_B + Fw_B - Wk_1w_Aw_B - Wk_2w_Bw_C \tag{4.27b}$$

$$0 = -Fw_C + 2Wk_1w_Aw_B - 2Wk_2w_Bw_C - k_3w_Cw_P \tag{4.27c}$$

$$0 = -Fw_E + 2Wk_2w_Bw_C \tag{4.27d}$$

$$0 = -Fw_P + Wk_2w_Bw_C - 0.5Wk_3w_Cw_P \tag{4.27e}$$

$$0 = -Fw_G + 1.5Ww_Cw_P \tag{4.27f}$$

in which, the state and measured variables $\bar{x}^s = \bar{y}^s = [w_A, w_B, w_C, w_E, w_P, w_G]^T$ are the mass fraction of the components; new scaled input variables are intro-

duced $\tilde{F}_B = F_B/10$ and $\tilde{T}_R = T_R/500$, so the manipulated variables $\tilde{\mathbf{u}} = [\tilde{F}_B, \tilde{T}_R]^T$ are the scaled feed flow rate of component B and the scaled reactor temperature; and, we consider the uncertain parameter $\tilde{\mathbf{d}} = [\tilde{F}_A]^T$ to be the scaled feed flow rate of component A , such that $\tilde{F}_A = F_A/10$, which is in fact an unmeasured disturbance. These new scaled variables are introduced only to equalize the magnitude order of the states, inputs, and parameters to enhance numerical aspects of controllability and observability. In addition, the spaces of the manipulated variables are defined as $F_B \in [1, 6]$ kg/s and $T_R \in [70, 100]$ °C.

The objective function of the problem, to be maximized, is a simple balance between the revenue with desired products and the cost with reactants:

$$J_{eco}(\tilde{\mathbf{y}}^s, \tilde{\mathbf{u}}, \tilde{\mathbf{d}}) = 1143.38w_P F + 25.92w_E F - 76.23F_A - 114.34F_B \quad (4.28)$$

Here we consider the presence of a single nonlinear constraint in component E as an upper bound:

$$w_E \leq 0.3 \quad (4.29)$$

To proceed with the linearization required by the proposed methodology, the Automatic Differentiation framework of CasADi was used (ANDERSSON *et al.*, 2019). The dynamic system was only used to produce plant measurements and, to do so, the CVODES algorithm from the SUNDIALS suite (HINDMARSH *et al.*, 2005) was used. The nonlinear programming solver IPOPT (WÄCHTER and BIEGLER, 2006) carried out the economic optimization through the interface of CasADi. In addition, the control problem was solved using the function “quadprog” from MATLAB®.

4.4.1 Hammerstein dynamic matrices identification

To proceed with the identification of the Hammerstein dynamic matrices, an input dynamic profile was imposed on the plant in a manner that each input variable was independently disturbed. This was carried out in both directions with an amplitude of ± 10 % of the nominal value, which was chosen as the optimal solution of the economic problem with the unmeasured disturbance $F_A = 1.8275$ kg/s. During the dynamics identification experiment, F_A was kept constant, and the measurements were free of noise. Figure 4.3 shows the input profile imposed on the plant. In addition, a sampling time of 1 min was considered.

A regression problem was formulated in a weighted least squares approach. Only the initial condition of the data window was used to predict the remainder of the dataset by the Hammerstein model. The unit weight was considered to

w_A , w_B , w_E and w_G , while the weights of w_C and w_P were considered 10 and 100, respectively. Special attention is given to the requirement of the stability of the estimated model and the absence of oscillatory behavior in the data by adding Equations 4.4 and 4.5 as constraints to the identification problem.

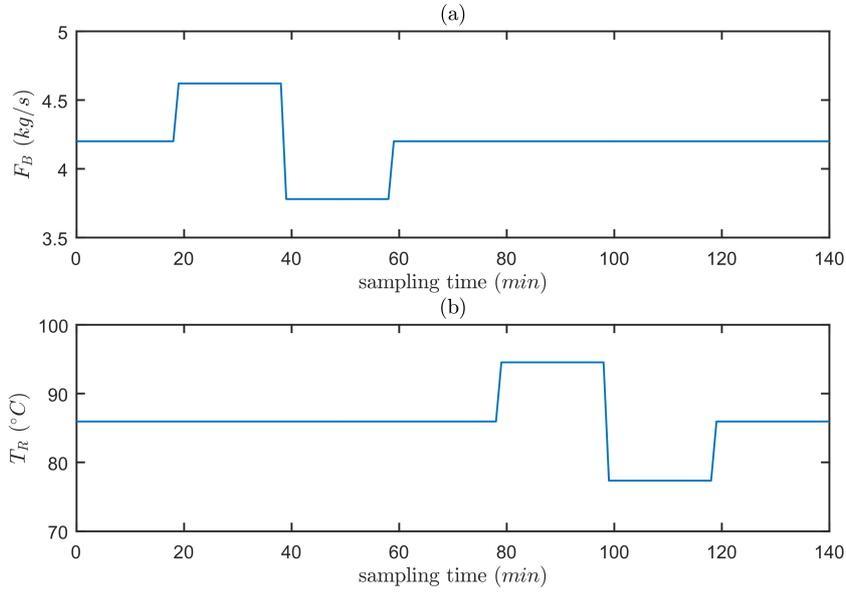


Figure 4.3: Input profile in the Hammerstein dynamic matrices identification: (a) F_B (kg/s); (b) T_R (°C).

Figure 4.4 shows the predictions of the output variables w_E and w_P made by the estimated Hammerstein models in the same identification dataset. Model 1 presents the worst behavior in terms of the dynamics even though it can match the steady state of the static model.

Visually, models 2 and 3 present very similar behavior, but the value of the identification objective function of model 3 is inferior to model 2; 0.0020 against 0.0033. Therefore, it seems that model 3 is a better local model than model 2 in the region of $\pm 10\%$ around the nominal input values. This is justified by the fact that it presents more parameters to be estimated. However, the higher number of parameters resulted only in a marginal benefit, almost imperceptible in a visual screening. Here, model 2 already hints at its superiority due to result in good performance despite its simplicity.

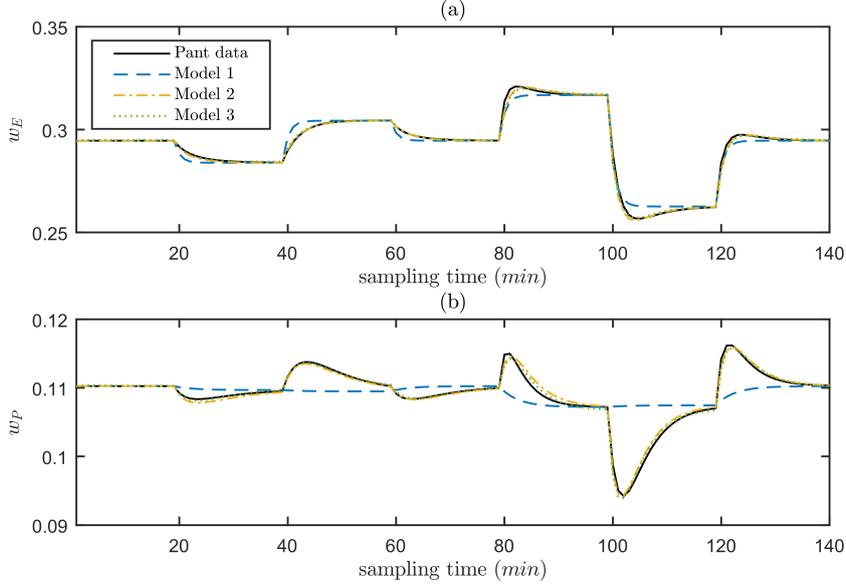


Figure 4.4: Output profile in the Hammerstein dynamic matrices identification: (a) w_E ; (b) w_P .

4.4.2 HEKF performance in open loop in the presence of measurement noise

Aiming to evaluate the performance of the proposed HEKF in terms of parameter estimation and measurement noise filtering, two filter tunings are proposed:

- Tuning 1: designed to prioritize the correction of the measurements over the model prediction by setting $\mathbf{Q} = 10^{-4}\mathbf{I}_{n_y \times n_d}$, $\mathbf{R} = 10^{-4}\mathbf{I}_{n_y}$ and initial covariance matrix as $\mathbf{P}_0^+ = 10^{-1}\mathbf{I}_{n_y \times n_d}$;
- Tuning 2: designed to prioritize the prediction of the model over correction of the measurements by setting $\mathbf{Q} = 10^{-4}\mathbf{I}_{n_y \times n_d}$, $\mathbf{R} = 3 \times 10^{-2}\mathbf{I}_{n_y}$ and $\mathbf{P}_0^+ = 10^{-1}\mathbf{I}_{n_y \times n_d}$.

The simulation scenario was designed in open loop, keeping the manipulated variables at their nominal values to minimize the influence of the controller dynamics in the estimation analysis. In this scenario, the true plant disturbance was drifted every 120 sampling instants in a pulse pattern with increasing amplitude from -5% to $+30\%$ of the nominal value. The measurements were built by corrupting the output variables with a Gaussian zero-mean noise with amplitude of 1% of the noise-free value. Figure 4.5 shows the comparison of the parameter estimation performance of the three proposed HEKF and a perfect model EKF in both tunings.

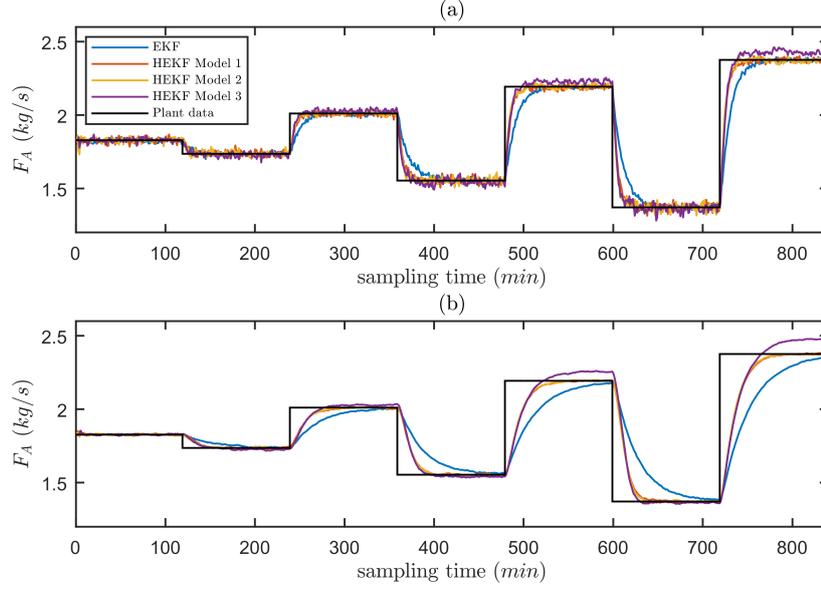


Figure 4.5: Disturbances estimation performance in open loop in the presence of measurement noise: (a) F_A (kg/s) estimation with tuning 1; (b) F_A (kg/s) estimation with tuning 2.

Apart from model 3, the HEKF performed similarly to the perfect model EKF, confirming that the proposed Hammerstein structures are valid as approximate dynamic models for estimation approaches, preserving the observability characteristics from the static model. It is possible to see that the parameter estimation performance of the HEKF using model 3 deteriorates as the operating point moves away from the identification point, which is undesirable in the context of RTO. This effect is expected since model 3 dynamic matrices are independent. Even though model 3 was observed as the best local model in the region of $\pm 10\%$ around the nominal input values, its behavior would be unsatisfactory far from this region or even with shifted unmeasured disturbances. Even with possible unrealistic dynamics in these scenarios, we can indeed state that at least the steady state of models 1 and 2 will match the static model. Provided the static model is reliable and reproduces well the equilibrium points of the plant, models 1 and 2 benefit from this reliability, unlike model 3.

The EKF presented the slowest dynamics compared to the other HEKFs, but this is only a matter of tuning and could be easily overcome by changing matrices Q and R . The performance of estimating the measured variables can be visualized in Figure 4.6 for tuning 1 and Figure 4.7 for tuning 2.

In all approaches, tuning 1 presented a fast estimation of the parameter to the real value of the plant. However, it is not able to properly attenuate the measurement noise, propagating it to the estimated quantities. On the other hand, tuning 2 presented more capability of filtering noise, but this comes with the cost of having

a slower parameter convergence to the true value. In addition, tuning 1 can correct the misled dynamics caused by an inaccurate dynamic model and the wrong value of the parameter during the convergence time as presented in Figure 4.6.

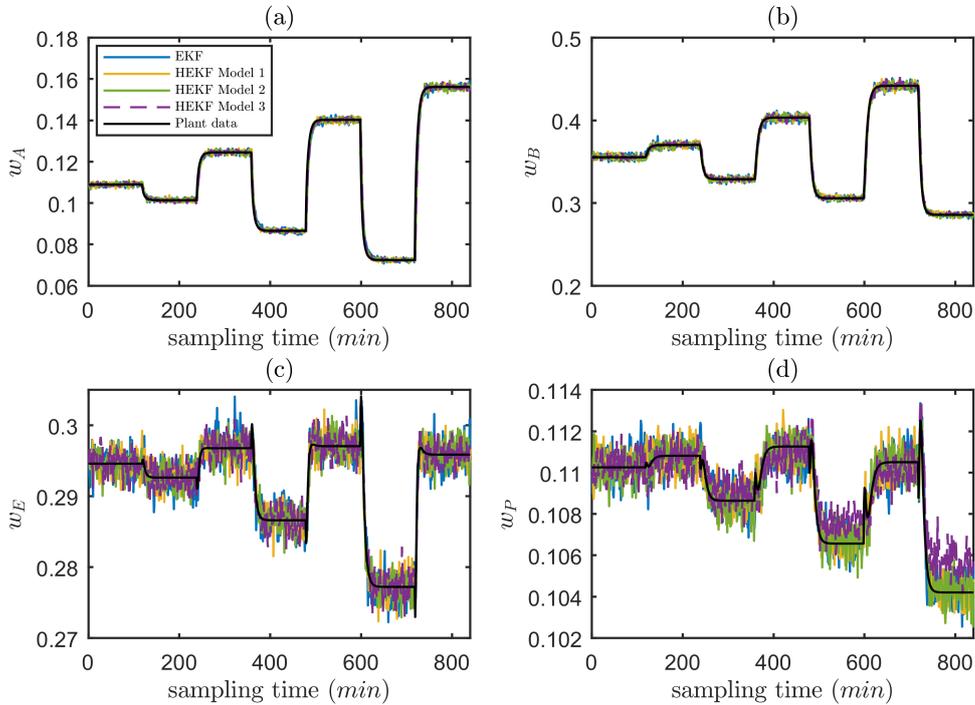


Figure 4.6: Output estimation performance with tuning 1 in open loop in the presence of measurement noise: (a) w_A ; (b) w_B ; (c) w_E ; (d) w_P .

HEKF using models 1 and 2 present very similar parameter estimation performance, even for disturbances that drive the conditions to regions far from the identification scenario. In every case, they were able to estimate the true value of the plant, indicating that this ability comes from the observability characteristics of the static model rather than the dynamics of linear ARX. However, considering HEKF using model 3, even though it presents a better fit to the process in the identification region, this strategy fails to estimate the true parameter value in regions far from the nominal operation, which can be observed by the offsets in the fourth and sixth steps of Figure 4.5. This effect is explained by the fact that model 3 does not guarantee to match the steady-state condition of the static model, resulting in the poor dynamic behavior of the estimated output variables in Figure 4.7.

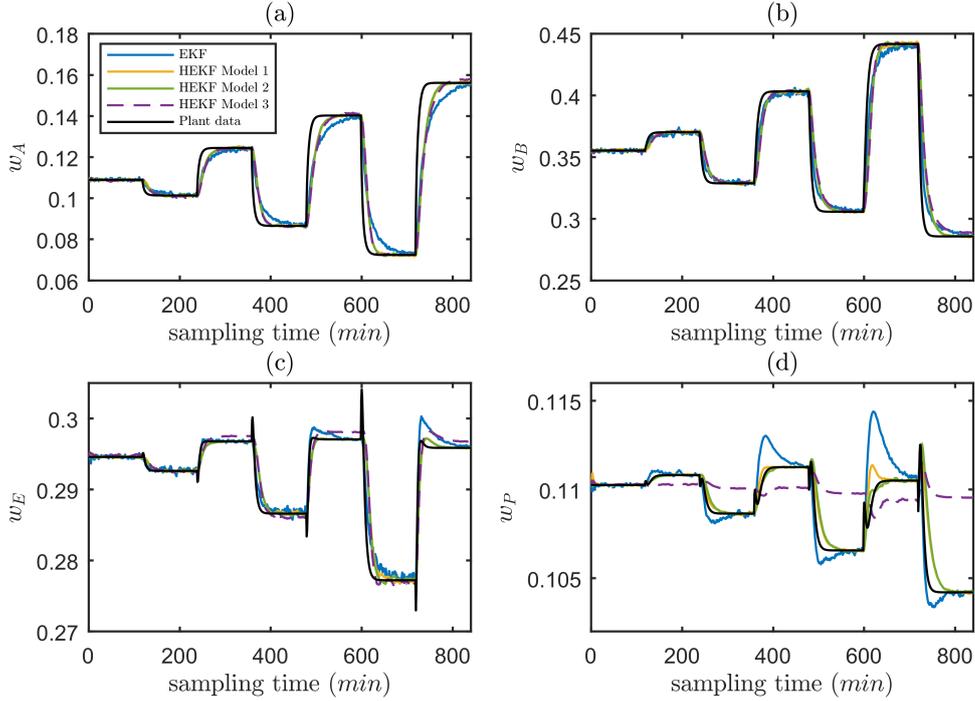


Figure 4.7: Output estimation performance with tuning 2 in open loop in the presence of measurement noise: (a) w_A ; (b) w_B ; (c) w_E ; (d) w_P .

Figure 4.7 shows that the slow parameter convergence can also result in the inaccurate dynamic behavior of the output variables. This effect added up with poor dynamic approximation led HEKF with model 3 to be the worst approach. Therefore, in the remainder of this chapter, model 3 was disregarded because it is considered inadequate in the sense of FORBES *et al.* (1994) for RTO purposes.

4.4.3 Closed-loop performance of the Adaptive Self-Optimizing Control Framework

The real-time adaptive SOC framework presented in Figure 4.2 was evaluated using the Hammerstein models 1 and 2 and compared to a framework similar to the proposal of GRACIANO *et al.* (2015). Besides our adaptive infinite-horizon MPC based on Hammerstein internal model, our framework differs from GRACIANO *et al.* (2015) in the RTO strategy using transient measurements, herein denoted as HRTO. GRACIANO *et al.* (2015) proposes the use of a static RTO, where a steady-state detection is followed by a static data reconciliation and parameter estimation approach, which estimated the unknown parameters based on static measurements. Here, we compare this approach to ours with the difference that we keep our adaptive controller in both approaches to accomplish a fairer comparison.

The simulation scenario comprises two-step perturbations in the unmeasured disturbances. The first was imposed in the 25th sampling instant with an ampli-

tude of +40 % of the nominal operation. The second was imposed in the 125th sampling instant with amplitude of -40 % of the nominal operation in a simulation window of 250 sampling instants. The second step generates an optimal solution with the constraint in component E active, while in the first step, there is no active constraint.

For all approaches, the same controller tuning is considered, the control horizon is 5 sampling instants, the weight matrices are $\mathbf{W}_c = \mathbf{I}_{n_u}$, $\mathbf{W}_{\Delta u} = 10^{-2}\mathbf{I}_{n_u}$ and the slack weight is 10 for each slack variable. These tuning parameters were chosen by trial experimentation since our aim was not to reach the best tuning under some performance criteria. However, for this goal, we refer to the work of GIRALDO *et al.* (2019). Since the effect of noise has been discussed in Section 4.4.2, in this section measurement noise was suppressed to clarify the effect of dynamics. However, all observations made in Section 4.4.2 would be made here if the noise were considered.

To illustrate the economical performance of each approach, not only the objective function is presented but also an accumulated loss function defined as:

$$L_k = \sum_{j=0}^k \left[J_{eco}(\bar{\mathbf{y}}_j^{sp}, \bar{\mathbf{u}}_j^*, \hat{\mathbf{d}}_{j-1}^+) - J_{eco,j}^* \right] \Delta t \quad (4.30)$$

in which $J_{eco,j}^*$ represents the true plant optimum at instant j and Δt is the sampling time, considered 60 s. It is important to mention that a reduction in the accumulated loss should be analyzed carefully. If the problem does not contain an active constraint, this reduction may be caused by a transient period that presents better economics than the desired steady-state setpoint. However, if a constraint is active, this reduction may be caused by its violation, which is undesirable. So, the reduction in the accumulated loss in the case of constraint violation cannot be interpreted as a positive economic aspect.

Figure 4.8 illustrates the performance of the manipulated variables and the estimation of the unmeasured disturbance in the self-optimizing HRT0, considering both HEKF tunings, compared to the self-optimizing RTO, considering the internal Hammerstein model 1.

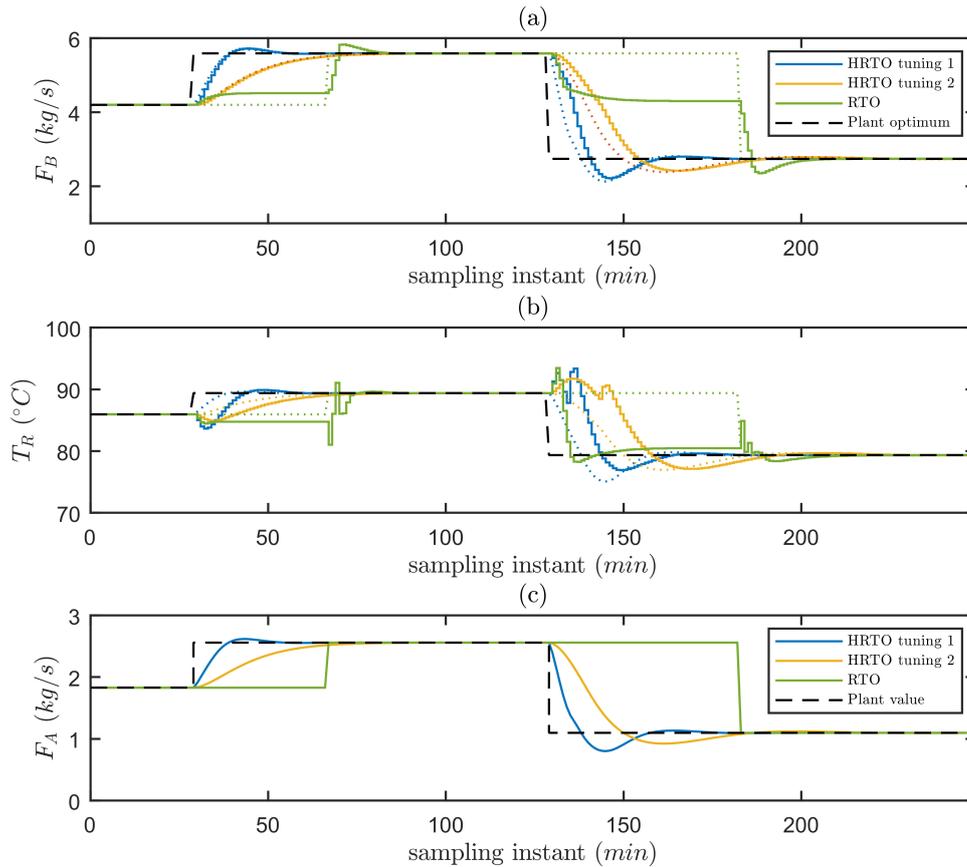


Figure 4.8: Closed-loop performance using Hammerstein model 1: (a) F_B (kg/s); (b) T_R (°C); (c) F_A (kg/s). The dotted lines represent the optimal inputs obtained by the HRTO.

In the HRTO approaches, the controller presents a soft response and reasonably follows the optimal input trajectory, even though these trajectories are not targets for the controller since the controller only tracks the self-optimizing variables. On the other hand, the RTO approach presented a slightly aggressive behavior, with the appearance of undesirable oscillatory moves in variable T_R at the moments that its setpoints are suddenly changed due to the steady-state detection. This effect could be avoided by increasing the IHMPC weights of the manipulated variables variations or by softening the parameter implementation using an exponential filter, for example. These solutions were not implemented because they would increase conservatism in the RTO approach and result in an even worse economic performance. The estimated performance of HEKF with tuning 1 and 2 followed the same conclusions drawn in Section 4.4.2 that tuning 1 presents a faster convergence, but it would present a worse noise filtering ability. Regarding the RTO approach, long periods of sub-optimal operation are observed due to steady-state waiting time. However, when it is detected, the estimator can

perfectly describe the actual value of the parameter, which is a fair behavior for practical applications if one considers the use of robust static M-estimators (DE MENEZES *et al.*, 2021). Figure 4.9 illustrates the output variables and the economic indexes for this scenario.

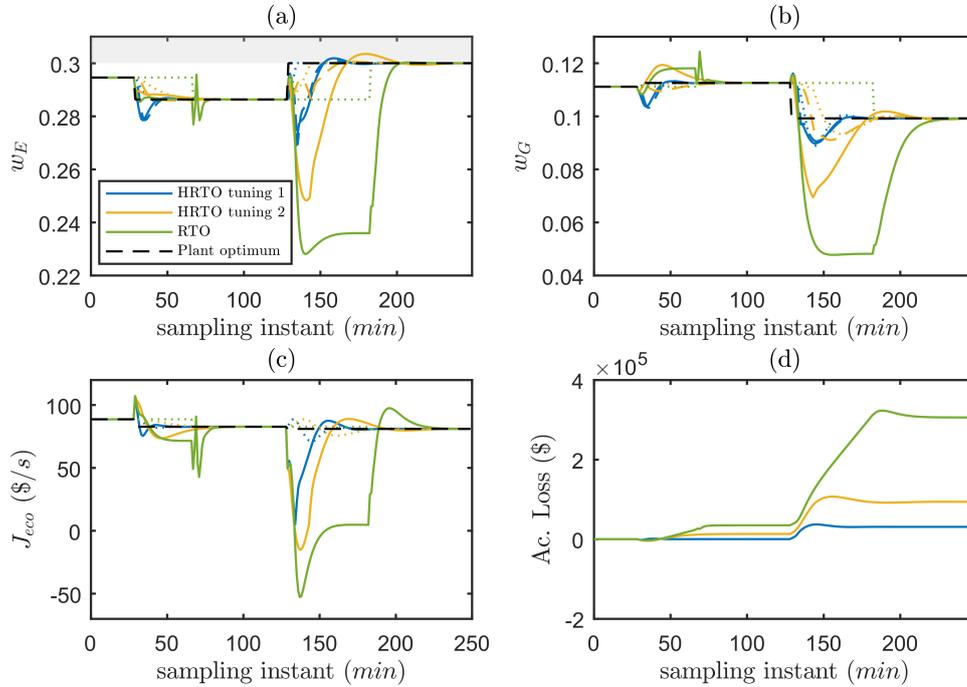


Figure 4.9: Closed-loop performance using Hammerstein model 1: (a) w_E ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function ($\$$). Dashed lines represent quantities estimated by the HEKF, dotted lines represent the optimal setpoints obtained by the HRTO and the grey area represents the violation of the imposed constraint.

All approaches can drive the system to its steady-state optimal operation. The main differences are observed in the transient windows, where the parameter is constantly updated in the HRTO, or the system is waiting for static data in the RTO approach.

Regarding the output variables estimation, it is clear that tuning 1 performs better than tuning 2. That is explained by the fact that model 1 is only a fair but not accurate approximation of the plant dynamics. Therefore, the measurement update plays an important role in correcting the dynamic trajectory of the output variables in this scenario, which directly impacts the superior economic performance of HRTO with tuning 1. The RTO approach presented the worst behavior since the system diverges significantly from the optimal operation in the steady-state waiting periods, which directly impacts the inferior economic performance of the RTO. Regarding the active constraints in the second step disturbances, only the HRTO approaches presented a period of a constraint violation but rapidly di-

recting the variable to its setpoint at the upper bound, which is a result that validates the methodology to handle changes in the active set of constraints presented in Section 4.3.4.

The performance of the proposed Hammerstein model 2 was also verified in the same simulation scenario. The manipulated variables and unmeasured disturbances can be visualized in Figure 4.10 and the output variables and economic indexes are illustrated in Figure 4.11.

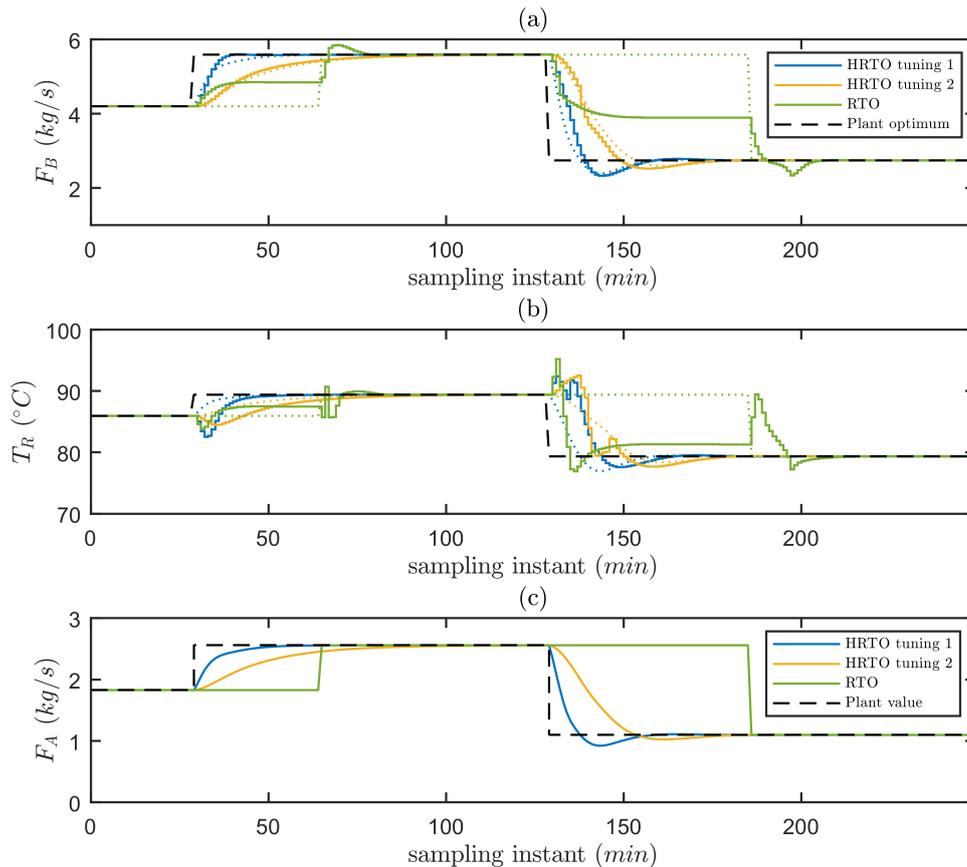


Figure 4.10: Closed-loop performance using Hammerstein model 2: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal setpoints obtained by the HRT0.

The remarks of the results with model 2 are very similar to model 1. However, comparing each other, it is possible to notice a superior performance of model 2. This is expected since model 2 better describes the plant dynamics.

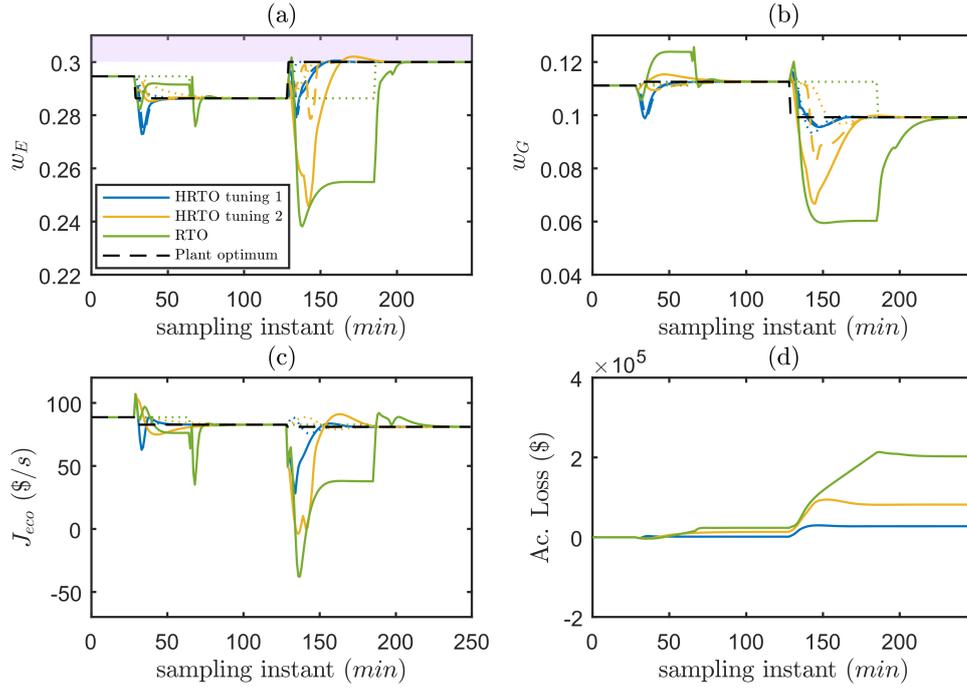


Figure 4.11: Closed-loop performance using Hammerstein model 2: (a) w_E ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function (\$). Dashed lines represent quantities estimated by the HEKF, dotted lines represent the optimal setpoints obtained by the HRTO and the grey area represents the violation of the imposed constraint.

Some approaches presented a slight violation of the constraint for a short period after the second disturbance. This violation occurred only in the plant data and not in the HEKF estimates, which is the variable used as the initial condition for the controller. The only exception occurred in the approach using model 1 and tuning 1, where both plant data and HEKF estimate violated active constraint during transient response. However, this violation does not necessarily imply using a slack variable to soften the constraint since the control uses a linearization of the Hammerstein model as the internal model and not a perfect dynamic model of the plant. In fact, the use of a slack variable depends on the tuning of the slack weights, as shown in Figure 4.12, for the weight used in the previous results $w_{w_E}^U = 10$ and for another tuning using $w_{w_E}^U = 10^{-5}$.

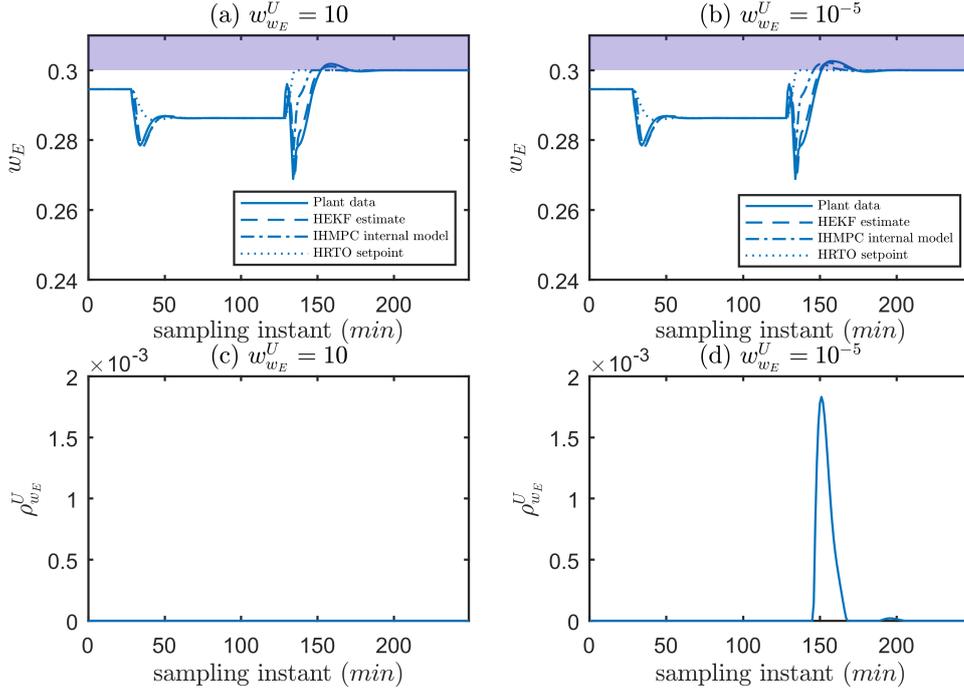


Figure 4.12: Evolution of the slack variable using Hammerstein model 1 and tuning 1: (a) w_E with $w_{w_E}^U = 10$; (b) w_E with $w_{w_E}^U = 10^{-5}$; (c) $\rho_{w_E}^U$ with $w_{w_E}^U = 10$; (d) $\rho_{w_E}^U$ with $w_{w_E}^U = 10^{-5}$.

The controller's internal model prediction does not violate the constraint despite the violation observed in the plant data and HEKF estimate for the proposed tuning of the slack weights. For a more relaxed tuning, such as the one presented in Figure 4.12 that considers $w_{w_E}^U = 10^{-5}$, it is possible to see that the controller's internal model prediction also violates the constraint. This is made possible by the activation of the slack variable related to the upper bound of w_E , which presents non-null values during the violation period. Despite both tunings showing a different behavior of the slack variable, this did not reflect a significant change in the controller's behavior.

To provide a better transverse comparison of the economic performances, Table 4.1 shows the terminal accumulated loss for each approach.

Table 4.1: Terminal accumulated terminal loss for each closed-loop approach

	HRTO tuning 1	HRTO tuning 2	RTO
model 1	0.31×10^5	0.94×10^5	3.01×10^5
model 2	0.28×10^5	0.82×10^5	2.03×10^5

Table 4.1 confirms that the proposed HRTO with self-optimizing adaptive IHMPC based on Hammerstein's approximate model outperformed the RTO approach based on the proposition of GRACIANO *et al.* (2015). Model 2 presents

better performance than model 1 due to a more accurate dynamic approximation of the plant. However, one must consider that model 2 requires n_y times more parameters to be identified than model 1. And, finally, tuning 1 presents better results than tuning 2. However, tuning 1 would have a worse filtering ability in the presence of noise, which would propagate noise for the estimated disturbances and parameters, generating noisy setpoints for the controller. The effect of noise is further discussed in the Supplementary Material of this chapter, Appendix C, where we present the closed-loop performance analysis of the frameworks in the presence of two measurement noise amplitudes, 1 % and 5 %.

Table 4.2 presents the average computational cost of each loop stage in terms of run time in ms . The algorithms ran in an Intel® Core™ i7-8565U 1.8GHz and 8GB DDR4 RAM memory.

Table 4.2: Performance of the proposed framework in terms of computational cost

Loop stage	Average iteration time (ms)	Time percentage (%)
Model Switching	1.29	2.7
HEKF	1.17	2.5
Optimization	22.41	47.4
SOC	3.38	7.2
IHMPC	17.02	36.0
Plant integration	1.96	4.2
Total Loop	47.23	100.0

It is interesting to note that the proposed framework is highly feasible to be applied during the 60 s sampling time. The average cost of the loop, including the plant integration, is 47.23 ms . It is worth mentioning that the limiting stages are the nonlinear optimization and the IHMPC, accounting for 47.4 % and 36.0 % of the total loop, respectively. In addition, the linearizations required in model switching and SOC stages are executed very efficiently using the automatic differentiation framework. Finally, comparing the computational cost of the RTO, the main difference is the presence of a static data reconciliation with simultaneous parameter estimation instead of the HEKF, which presented an average iteration time of 44.51 ms . Although the RTO only runs when a steady state is detected, its computational cost is considerably higher than the proposed HRTO scheme due to the presence of static reconciliation and estimation.

As already mentioned, the main advantage of the proposed approach is to incorporate information from the static nonlinear model into the models used in the control layer. Depending on how this is done, it is possible to match the

steady-state condition of the optimization and control layers, such as in models 1 and 2. However, it is essential to point out the limitations associated with linear dynamic approximation. For systems with accentuated nonlinear dynamics, the proposed model may fail to describe the system satisfactorily during transient periods, even in operating points near the identification region. Furthermore, not only the proposed approach but also the original HRTO, in general, should be deeper investigated in different classes of nonlinear systems. For instance, it has been shown that unknown dead times could significantly degenerate HRTO performance (CURVELO *et al.*, 2021).

4.5 Conclusions

This chapter presents a new HRTO framework considering the unavailability of a mechanistic dynamic model of the process. A Hammerstein structure based on the available static model is proposed to work as an approximate dynamic model. An Extended Kalman Filter and a Self-Optimizing Infinite Horizon MPC are formulated based on this structure. The proposal was validated in the Williams-Otto reactor in the presence of parametric uncertainty. Three models were proposed based on the Hammerstein structure. The main conclusions are:

- It was found that model 3 is not adequate for the HRTO considered fixed dynamic matrices. On the other hand, models 1 and 2 presented great performance due to the property that the approximate dynamic models match the steady state of the static model;
- Although model 2 presented higher accuracy than model 1, it has the squared number of parameters of model 1 to be estimated, which can be concerning considering a large-scale system;
- The open-loop results showed that the HEKF can preserve the observability characteristics of the static model, performing similarly to the perfect model EKF. In addition, the HEKF presented a compromise between fast parameter estimation and rejecting measurement noise;
- Regarding the closed-loop results, the proposed framework outperformed the framework that uses SSD in a classic RTO formulation. Although both frameworks presented the same ability to drive the plant to its optimum at a steady state;
- Regardless of the HEKF tuning, the proposed framework presented improved economics, especially in a scenario of change in the active constraints.

Finally, although the framework was only tested in a benchmark system, the results of the computational cost of each stage of the proposed framework show that it has the potential to be applied in industrial-scale systems. Therefore, this is a matter of interest for future investigation.

Chapter 5

Steady-state real-time optimization using transient measurements and approximated Hammerstein dynamic model: A proof of concept in an experimental rig

A version of this chapter was published in the Journal of Process Control, Volume 132, December 2023 (DELOU *et al.*, 2023b).

5.1 Introduction

Focused on removing the rigorous dynamic model availability requirement and improving ROPA's applicability, in Chapter 4, we proposed a specific Hammerstein dynamic model approximation that takes advantage of the available static model on its structure. The proposed approximate model combines the static model with virtual linear dynamics introduced by an ARX. The ARX structure introduces a matrix of dynamic parameters that should be identified from plant data, which is named here as the dynamic matrix. In this new approach, called Hammerstein ROPA (HROPA), the Hammerstein dynamic model replaces the rigorous dynamic model in the model adaptation stage. In a simulated environment, this structure was shown to have the capability of preserving the model parameters and disturbance observability; it is naturally compatible with the static model as it always matches its steady-state conditions, and the linear dynamic approximation can be fairly accurate depending on the system in study. However, previous studies were limited to the application of the HROPA methodology in

a simulated environment, where a rigorous dynamic model was used to emulate the plant responses, and all parametric or structural plant-model mismatches are known. The main challenge of the current work is to implement the previously proposed HROPA methodology in an experimental plant, where a perfect model of the plant is not available and, therefore, unknown plant-model mismatches are definitely present. Besides, this was carried out on the same experimental setup used in MATIAS *et al.* (2022), allowing a comprehensive comparison among the studied RTO approaches.

The main contribution of this work is in the context of the previously proposed Hammerstein structure by implementing an HROPA strategy to a lab-scale experimental rig. The study intends to show the potential of the methodology to work with real systems and increase the applicability of ROPA approaches toward large-scale systems. The published version of this chapter also included a thorough literature review of the use of transient measurement within the static RTO framework, adapted from Section 2.6.

This chapter implements two versions of HROPA in a small-scale experimental rig that emulates a subsea oil well network. The first version considers the originally proposed Hammerstein model, where a *a priori* identification step based on plant data takes place. The second explores the direct use of the static model in the EKF algorithm, making the static model work as an instantaneous dynamic model. This second version was explored since the studied system presented rather fast responses. The performance of HROPA is compared with the ROPA implementation, which was carried out by MATIAS *et al.* (2022) on the same experimental rig under the same disturbance scenario. It is shown that HROPA presented similar operational profiles to the classic ROPA with a slightly lower economic performance. Although it is expected that HROPA presents a slightly lower economic performance compared with ROPA, our goal is to show the proof of concept of the HROPA methodology and to collaborate to increase the industrial applicability of ROPA-like algorithms since the HROPA methodology can be applied without having a rigorous dynamic model of the plant.

The main contributions of the present chapter are:

- General guidelines for the Hammerstein dynamic matrix identification for practical implementation;
- Proof of concept of the previously proposed HROPA approach in a lab-scale experiment rig;
- Use of the static model directly in the dynamic model adaptation algorithm.

The remainder of the chapter is organized as follows. Section 5.2 presents the ROPA using a dynamic Hammerstein model approximation, guidelines for the

dynamic matrix identification, the formulation of the dynamic parameter estimation, and steady-state economic optimization. Section 5.3 provides the details of the experimental rig used as a case study, the experimental layout, the available process model, the optimization objective and constraints, the Hammerstein model identification scenario, the closed-loop disturbance scenario, and the EKF tuning details. Section 5.4 exhibits the results and discussion comparing the two HROPA approaches with the originally proposed ROPA method. Finally, Section 5.5 summarizes the conclusion of this work.

5.2 ROPA using Hammerstein dynamic model approximation

The ROPA algorithm differs from the classic RTO, mainly in how measured data is incorporated into the framework. As classic RTO adapts the static model relying on static parameter estimation, it is mandatory to use only steady-state measurement data. Therefore, a steady-state detection strategy is used to monitor measurements and decide whether or not to run the RTO framework, generating long waiting times. In fact, this has been reported as one of the major RTO issues (DARBY *et al.*, 2011; FRIEDMAN, 1995). On the other hand, ROPA replaces the static parameter estimation with a dynamic one, such as Kalman-filter type observers. This simple modification removes the necessity for the steady-state detection stage, enabling virtually any kind of measurement to be incorporated into the optimization framework, regardless of its dynamic or static nature. However, this simple idea also creates a previously nonexistent problem in RTO projects, which is the requirement of having a dynamic mechanistic model, not only reliable but also compatible with the static model used for optimization. Such an issue must not be unconsidered as it could significantly impact the applicability of ROPA algorithms in real process plants and even hinder the project payoff.

To tackle this issue, DELOU *et al.* (2021c) proposed the Hammerstein ROPA (HROPA), which is an adaptation of the originally proposed ROPA approach by replacing the rigorous dynamic model with an approximate Hammerstein model. This Hammerstein model is built by combining the available static model with a linear ARX dynamic approximation. The framework differences between RTO, ROPA, and HROPA can be visualized in Figure 5.1.

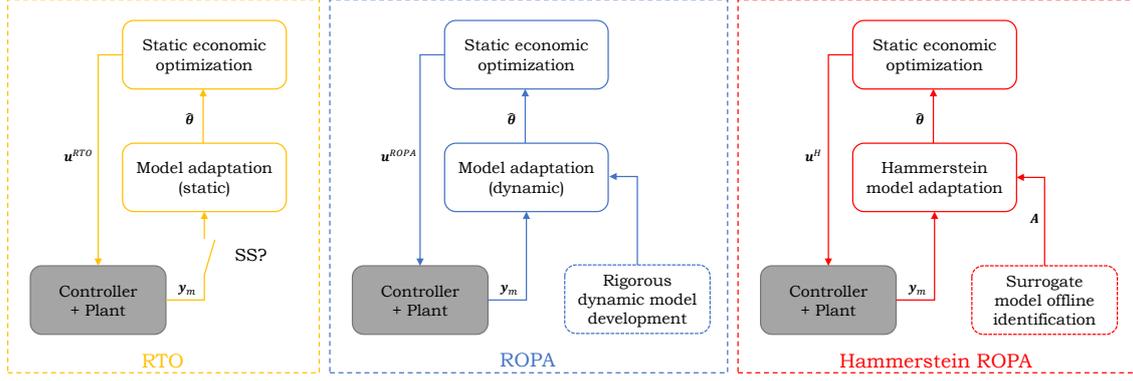


Figure 5.1: Block diagram comparing classic RTO, ROPA, and Hammerstein ROPA.

Both ROPA and Hammerstein ROPA approaches include an extra step to the RTO project, which is the development of a rigorous and approximate dynamic model, respectively. However, the effort required by identifying the Hammerstein model is considerably lower than developing a mechanistic dynamic model, especially considering large-scale systems. Moreover, identifying the Hammerstein dynamic matrix is similar to the procedures carried out in linear MPC projects. Considering that a linear MPC internal model might be frequently available, the Hammerstein dynamic matrix can be derived from the controller internal model to minimize interventions and experiments imposed directly on the plant. Therefore, the HROPA algorithm presents a higher potential applicability to large-scale systems than the originally proposed ROPA.

5.2.1 Hammerstein model structure

A general static model can be represented in its state space formulation:

$$\mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \quad (5.1a)$$

$$\mathbf{y} = \mathcal{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \quad (5.1b)$$

in which, $\mathbf{x} \in \mathbb{R}^{n_x}$ represents the state variables vector, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the input variables vector, $\mathbf{y} \in \mathbb{R}^{n_y}$ represents the measured outputs vector, and $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ represents the vector of uncertain parameters and disturbances. Functions $\mathcal{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x}$ and $\mathcal{H} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_y}$ represent, respectively, the steady-state mapping of \mathbf{x} and \mathbf{y} in respect to \mathbf{u} and $\boldsymbol{\theta}$. This model can be simplified without loss of generality as:

$$\mathbf{y} = \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}) \quad (5.2)$$

The Hammerstein structure proposed by DELOU *et al.* (2021c) consists of using the above nonlinear mapping to process the inputs and parameters signal and then a feedback linear ARX model to introduce virtual dynamics. The Hammerstein model is presented below.

$$\mathbf{y}_{k+1}^h = \mathbf{A}^h \mathbf{y}_k^h + (\mathbf{I} - \mathbf{A}^h) \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}_k) \quad (5.3)$$

in which \mathbf{y}_k^h represents the measured vector predicted by the Hammerstein model at instant k . Matrix $\mathbf{A}^h \in \mathbb{R}^{n_y \times n_y}$ is the dynamic matrix. This matrix should be previously identified in an offline identification step, which is further discussed in the next subsection.

Here, two conditions that any approximate dynamic model should respect to be considered adequate to be used in a ROPA framework are outlined:

1. (Compatibility condition) If \mathbf{u} is kept constant at \mathbf{u}_∞ and $\boldsymbol{\theta}$ does not change in time, the steady states of the dynamic approximation and the static model must match: $\lim_{k \rightarrow \infty} \mathbf{y}_k^h = \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})$;
2. (Observability condition) The uncertain parameters and disturbances from the static model must be observable given the available plant measurements.

The compatibility condition can be met by Equation 5.3 under the assumption that the dynamic matrix \mathbf{A}^h presents the property to be stable. This assumption is fairly easy to enforce during the identification of the dynamic matrix. This condition can be easily visualized by writing Equation 5.3 in the following manner:

$$\bar{\mathbf{y}}_{k+1}^h = \mathbf{A}^h \bar{\mathbf{y}}_k^h \quad (5.4)$$

in which $\bar{\mathbf{y}}_k^h := \mathbf{y}_k^h - \mathbf{y}(\mathbf{u}_k, \boldsymbol{\theta}_k)$.

The observability condition is met by Equation 5.3 since it explicitly uses the static model in its formulation, assuming that the uncertain parameters and disturbances are identifiable via the static model; this fact was also shown in practice by DELOU *et al.* (2021c).

Beyond those two conditions, another critical factor to consider is how accurate the Hammerstein model can be to represent the actual system dynamics. Considering the linear dynamics presented in Equation 5.3, it is clear that the model's accuracy will remain in a local region around the identification reference point. To overcome this issue, adaptive schemes, higher-order or nonlinear model structures could be used. However, it is noteworthy that inaccurate dynamics will deteriorate the ROPA during the dynamic transitions as a consequence of inaccurate

parameter and disturbance estimation, but it will not prevent the convergence towards the true parameter values considering that the compatibility condition is met, as shown in DELOU *et al.* (2021b).

5.2.2 Identification of the dynamic matrix

The identification of the dynamic matrix is proposed to be carried out offline, similarly to the identification of linear dynamic models in classic MPC projects. If such models are available, it is possible to identify the Hammerstein model based on a controller internal model. However, here we assume that this is not the case.

There are many ways to carry out the identification procedure (NAJIM and IKONEN, 2001). This work proposes selecting a common operational condition to be used as a reference and imposing positive and negative disturbances around this reference independently for each input variable. The goal is to obtain a sufficiently excited data set for the identification procedure. The set of input data is represented by $\mathbf{U}^{id} \in \mathbb{R}^{N_{id} \times n_u}$, where N_{id} is the number of time samples in the identification set. It is reasonable to assume that the static model uncertain parameters and disturbances, $\Theta^{id} \in \mathbb{R}^{N_{id} \times n_\theta}$, are known during the dynamic matrix identification. After imposing the input data set to the plant, this will result in a measured output data set, represented by $\mathbf{Y}^{id} \in \mathbb{R}^{(N_{id}+1) \times n_y}$.

With the identification data sets in hand, the goal of the identification procedure will be to produce a dynamic matrix \mathbf{A}^h such that it minimizes the distance between the measured output data set from the Hammerstein model prediction, by taking the initial condition from the data and propagating it throughout the whole identification horizon. However, to account for the compatibility condition, which states that the resulting Hammerstein model must be stable, it is important to include the following nonlinear constraint:

$$\|\boldsymbol{\lambda}(\mathbf{A}^h)\| < 1 \quad (5.5)$$

in which, $\boldsymbol{\lambda}(\mathbf{A}^h)$ represents the eigenvalues of matrix \mathbf{A}^h . In addition, it might also be the case that one should avoid the rise of oscillatory modes in the resulting Hammerstein model if the original process does not present oscillations. This can be guaranteed by including the following constraint:

$$\text{Im}(\boldsymbol{\lambda}(\mathbf{A}^h)) = 0 \quad (5.6)$$

Therefore, the identification optimization problem can be written as the fol-

lowing weighted least square estimation:

$$\mathbf{A}^h = \arg \min_{\mathbf{A}^h} J_{id} := \sum_{k=1}^{N_{id}+1} (\mathbf{y}_k^{id} - \mathbf{y}_k^h)^T \mathbf{W}_{id} (\mathbf{y}_k^{id} - \mathbf{y}_k^h) \quad (5.7a)$$

$$\text{s.t. } \mathbf{y}_{k+1}^h = \mathbf{A}^h \mathbf{y}_k^h + (\mathbf{I} - \mathbf{A}^h) \mathbf{y}(\mathbf{u}_k^{id}, \boldsymbol{\theta}_k^{id}), \quad k = \{0, \dots, N_{id}\} \quad (5.7b)$$

$$\mathbf{y}_0^h = \mathbf{y}_0^{id} \quad (5.7c)$$

$$\|\boldsymbol{\lambda}(\mathbf{A}^h)\| < 1 \quad (5.7d)$$

$$\text{Im}(\boldsymbol{\lambda}(\mathbf{A}^h)) = 0 \quad (5.7e)$$

in which $\mathbf{W}_{id} \in \mathbb{R}^{n_y \times n_y}$ is a squared definite positive matrix of weights, which can be arbitrarily selected or selected as the inverse of the covariance matrix.

Although the inclusion of constraints 5.7d and 5.7e are required to meet the compatibility condition and the accuracy condition, respectively, it also includes a non-convexity into the identification problem.

5.2.3 Dynamic parameter estimation

The dynamic estimation strategy is the main core that differentiates the ROPA from the classic RTO frameworks, which is the building block that enables the use of dynamic data. In principle, among recursive or optimization-based strategies, any dynamic estimation approach would be suitable (VALLURU and PATWARDHAN, 2019). The EKF is an example of a recursive approach (SIMON, 2006b), while the MHE is an example of an optimization-based approach (PATWARDHAN *et al.*, 2012). In selecting which strategy to use, it is recommended to analyze the trade-off between estimation quality and computational cost of each method. The EKF is a compromise option since it presents a fairly accurate estimation with a significantly low computational cost.

DELOU *et al.* (2021c) proposed the formulation of the HEKF that incorporates the Hammerstein model presented in Equation 5.3 into the EKF framework in a simultaneous state and parameter estimation approach. The algorithm estimates the uncertain parameters at any instant k , by assuming instantaneous dynamics. This estimation is used to adapt the model for steady-state economic optimization. We refer to DELOU *et al.* (2021c) for the complete formulation.

The EKF tuning is a fundamental stage that must not be overlooked. A badly tuned EKF could hinder the operation and possibly destabilize the system (MATIAS and LE ROUX, 2018). However, the task of selecting suitable covariance matrices is not a simple thing. In addition, a systematic EKF tuning methodology is still an open topic in the literature. In this sense, it is noteworthy that the effort to tune the HEKF is significantly less than to tune an EKF that relies on a dynamic

mechanistic model. This is observed by the fact that the state variable dimension is n_y for the HEKF, while it is n_x for the EKF, and usually $n_y \ll n_x$. Therefore, HEKF presents significantly fewer tuning parameters compared to EKF.

5.2.4 Steady-state economic optimization

The following steady-state economic optimization is calculated on every ROPA iteration to determine the optimal set of inputs.

$$\mathbf{u}_{k+1}^* = \arg \min_{\mathbf{u}, \mathbf{x}} J_{ec}(\mathbf{y}, \mathbf{u}, \hat{\boldsymbol{\theta}}_k^+) \quad (5.8a)$$

$$\text{s.t. } \mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}, \hat{\boldsymbol{\theta}}_k^+) \quad (5.8b)$$

$$\mathbf{y} = \mathcal{H}(\mathbf{x}, \mathbf{u}, \hat{\boldsymbol{\theta}}_k^+) \quad (5.8c)$$

$$\mathcal{G}(\mathbf{x}, \mathbf{u}, \hat{\boldsymbol{\theta}}_k^+) \leq \mathbf{0} \quad (5.8d)$$

$$\mathbf{y}^{lb} \leq \mathbf{y} \leq \mathbf{y}^{ub} \quad (5.8e)$$

$$\mathbf{u}^{lb} \leq \mathbf{u} \leq \mathbf{u}^{ub} \quad (5.8f)$$

in which J_{ec} is the economic objective function of the optimization problem, $\hat{\boldsymbol{\theta}}_k^+ \in \mathbb{R}^{n_\theta}$ is the EKF *a posteriori* estimation of the uncertain parameters at instant k , $\mathcal{H} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_h}$ is the function that maps any additional nonlinear constraint that the problem may be subjected, and the superscripts $(\cdot)^{lb}$ and $(\cdot)^{ub}$ denotes lower and upper bounds, respectively.

5.2.5 Algorithmic implementation

This section presents the algorithmic implementation of ROPA as presented in MATIAS *et al.* (2022) and the Hammerstein ROPA. The first can be visualized in Algorithm 2, and the second in Algorithm 3. This representation enables a better comprehension of how each step of the ROPA and HROPA architectures are implemented in real time, in which the main difference relies on the fact that the first uses a rigorous high-fidelity dynamic model, while the second uses a dynamic Hammerstein model, which combines an approximate ARX dynamic with a rigorous high-fidelity static model.

Algorithm 2: ROPA with rigorous dynamic model (MATIAS *et al.*, 2022).

At every ROPA execution cycle k :

1. Acquire output and input plant measurements, \mathbf{y}_k^m and \mathbf{u}_k^m

Dynamic model adaptation (EKF):

2. Evaluate *a priori* augmented system states from rigorous dynamic model and augmented covariance matrix
3. Compute system sensitivity matrices from rigorous dynamic model: augmented state transition matrix and observation matrix
4. Compute Kalman gain and *a posteriori* augmented system states and augmented covariance matrix
5. Obtain updated parameter vector, $\hat{\boldsymbol{\theta}}_k^+$ from augmented system states

Steady-state economic optimization:

6. Update static model: $\mathbf{0} = \mathcal{F}(\cdot, \cdot, \hat{\boldsymbol{\theta}}_k^+)$ and $\mathbf{y} = \mathcal{H}(\cdot, \cdot, \hat{\boldsymbol{\theta}}_k^+)$
 7. Compute \mathbf{u}_{k+1}^* using Equation 5.8
 8. Apply input filter $\mathbf{u}_{k+1} = \mathbf{u}_k^m + \mathbf{K}_u (\mathbf{u}_{k+1}^* - \mathbf{u}_k^m)$
 9. Implement \mathbf{u}_{k+1} to the plant
-

Algorithm 3: Hammerstein ROPA. For the detailed implementation of the Hammerstein EKF, we refer to DELOU *et al.* (2021c).

At every HROPA execution cycle k :

1. Acquire output and input plant measurements, \mathbf{y}_k^m and \mathbf{u}_k^m

Dynamic model adaptation (EKF):

2. Evaluate *a priori* augmented system states from Hammerstein model and augmented covariance matrix
3. Compute the steady-state sensitivity matrix of the outputs in relation to the uncertain parameters, and derive the augmented state transition matrix and observation matrix
4. Compute Kalman gain and *a posteriori* augmented system states and augmented covariance matrix
5. Obtain updated parameter vector, $\hat{\boldsymbol{\theta}}_k^+$ from augmented system states

Steady-state economic optimization:

6. Update static model: $\mathbf{0} = \mathcal{F}(\cdot, \cdot, \hat{\boldsymbol{\theta}}_k^+)$ and $\mathbf{y} = \mathcal{H}(\cdot, \cdot, \hat{\boldsymbol{\theta}}_k^+)$
 7. Compute \mathbf{u}_{k+1}^* using Equation 5.8
 8. Apply input filter $\mathbf{u}_{k+1} = \mathbf{u}_k^m + \mathbf{K}_u (\mathbf{u}_{k+1}^* - \mathbf{u}_k^m)$
 9. Implement \mathbf{u}_{k+1} to the plant
-

5.3 Case study: Experimental rig

The experimental rig used in the current work is installed at the Department of Chemical Engineering of the NTNU and has been described in MATIAS *et al.* (2022). The purpose of the equipment is to reproduce some of the physics of a sub-sea oil well network on small scale. To be more specific, the experiment reproduces an artificial lift effect caused by a gas stream injected into a liquid well, which is observed in a gas-lifted system. In this kind of system, the gas injection results in a reduced liquid density and contributes to increasing oil production. However, increasing the gas flow rate is beneficial up to a certain level. There is a critical point where, if the gas flow rate continues to increase, the effect of frictional pressure drop dominates the density reduction and the oil production decreases (MATIAS *et al.*, 2022). Therefore, an optimal gas flow rate can be achieved, which can be determined by RTO strategies.

5.3.1 Experiment layout

The schematic of the rig can be visualized in Figure 5.2

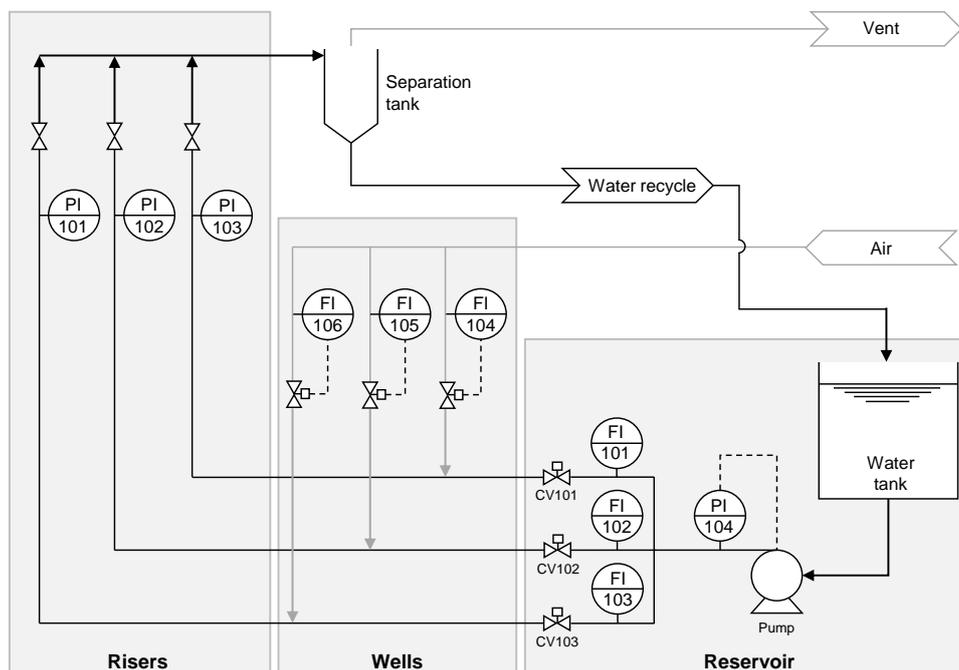


Figure 5.2: Experimental rig schematic adapted from MATIAS *et al.* (2022).

The rig can be subdivided into 3 main sections: the reservoir, wells, and risers. In the reservoir section, the main elements are a 200 L water tank that represents the actual reservoir, a centrifugal pump with discharge pressure control, and three control valves. The choice of water as the liquid fluid was made for simplification

purposes. Water is easier to handle and more environmentally friendly than oil and the gas lift effect, which is the phenomenon of interest, is still observable in water. It is also important to mention that this setup only copes with water production. Therefore no oil, water, and gas mixture phenomena are explored. Due to the pump configuration, the total reservoir production per well is limited between [2, 15] L/min. The three control valves can be manipulated to represent different well behaviors, for example, closing the valve during the experiment emulates well depletion. Individual well flow rates can be measured by flow meters installed upstream of the control valves.

The wells section consists of the gas injection points downstream to the reservoir control valves. The selected gas is air, again for the sake of simplicity, and its injection rate is controlled by air flow rate controllers. The air flow meters are installed upstream to the air injection valves, and the allowed air flow rate range is between [1, 5] slm per well.

The riser section comprises three vertical pipelines with 2.2 m of height. The risers are perpendicular to the well section, and there is a pressure meter at the top of each one. Upstream to the meters, there are manual valves, which are only used for system maintenance and maneuverability and are kept fully open during the experiment. The air-water mixture is sent to a separation tank, where air is vented to the atmosphere, and the water is recycled to the reservoir water tank to avoid waste.

5.3.2 Process model

The experimental rig dynamic model is detailed in MATIAS *et al.* (2022). However, in this work, we consider it unavailable, while its static counterpart is considered available. The static model set of equations is presented below:

$$0 = -w_{g,i} + w_{g,i}^{out} \quad (5.9a)$$

$$0 = -w_{l,i} + w_{l,i}^{out} \quad (5.9b)$$

$$0 = -w_{total,i} + w_{g,i}^{out} + w_{l,i}^{out} \quad (5.9c)$$

$$0 = -w_{l,i} + v_{o,i}\theta_{res,i}\sqrt{\rho_{l,i}(P_{pump} - P_{u,i})} \quad (5.9d)$$

$$0 = -P_{u,i} + P_{r,i} + \rho_{mix,i}g\Delta h_i + \frac{128\rho_{mix,i}(w_{g,i} + w_{l,i})L_i}{\pi\rho_{mix,i}D_i^4} \quad (5.9e)$$

$$0 = -\rho_{mix,i} + \frac{m_{g,i} + m_{l,i}}{V_{total,i}} \quad (5.9f)$$

$$0 = -V_{total,i} + \frac{m_{l,i}}{\rho_{l,i}} + \frac{m_{g,i}}{\rho_{g,i}} \quad (5.9g)$$

$$0 = -\rho_{g,i} + \frac{P_{u,i}M_g}{RT} \quad (5.9h)$$

$$0 = -w_{total,i} + \theta_{top,i} \sqrt{\rho_{mix,i}(P_{r,i} - P_{atm})} \quad (5.9i)$$

$$0 = -\frac{m_{l,i}}{m_{l,i} + m_{g,i}} + \frac{w_{l,i}^{out}}{w_{total,i}} \quad (5.9j)$$

where, the subscript i refers to any individual well from a set of $\mathbb{W} := \mathbb{I}_{1:n_w}$, with $n_w = 3$ in our case. The variables $m_{g,i}$ and $m_{l,i}$ represent gas and liquid holdups, respectively. $w_{l,i}$ is the liquid injection flow rate. $w_{g,i}^{out}$ and $w_{l,i}^{out}$ are the gas and liquid production flow rate, respectively. $P_{r,i}$ is the pressure at the riser head, $P_{u,i}$ is the pressure upstream to the injection point, P_{pump} is the pump outlet pressure, $\rho_{mix,i}$ is the riser fluid density, and $\rho_{g,i}$ the gas density.

The modeling hypotheses are discussed in detailed in MATIAS *et al.* (2022). The main simplifications are: (1) in Equation 5.9e, the pressure drop between the riser head and the injection point depends on the static fluid pressure and the pressure drop due to friction. The former is computed using the Darcy-Weisbach equation for laminar flows (last term of Equation 5.9e); (2) the liquid flowrate fraction is proportional to the liquid holdup in the pipes (Equation 5.9j).

The system presents $n_x = 30$ state variables in total. To obtain a more compact representation, we combined the variables that represent the same physical quantities in the individual wells using a bold notation. The resulting system state vector is:

$$\mathbf{x} := [\mathbf{m}_g^T, \mathbf{m}_l^T, \mathbf{w}_{total}^T, \mathbf{w}_l^T, (\mathbf{w}_g^{out})^T, (\mathbf{w}_l^{out})^T, \mathbf{P}_r^T, \mathbf{P}_u^T, \boldsymbol{\rho}_{mix}^T, \boldsymbol{\rho}_g^T]^T \quad (5.10)$$

The input variable vector is composed of $n_u = 7$ variables:

$$\mathbf{u} := [\mathbf{w}_g^T, \mathbf{v}_o^T, P_{pump}]^T \quad (5.11)$$

Note that, since the three wells share the same pump, we have only one pump pressure outlet in the input vector.

The measured variables are the liquid volumetric flow rate $\mathbf{Q}_l \in \mathbb{R}^{n_w}$ and the riser head pressure, $\mathbf{P}_r \in \mathbb{R}^{n_w}$. The set of equations that define the measurement predictions is:

$$\begin{aligned} \mathbf{y} &:= [\mathbf{Q}_l^T, \mathbf{P}_r^T]^T \\ &= \mathbf{H}\mathbf{x} \\ &= \begin{bmatrix} \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{I}_{n_w}/\rho_l & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} \\ \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{I}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} & \mathbf{0}_{n_w} \end{bmatrix} \mathbf{x} \end{aligned} \quad (5.12)$$

Finally, the reservoir valve flow coefficients, $\theta_{res} \in \mathbb{R}^{n_w}$, and the top valve flow coefficient, $\theta_{top} \in \mathbb{R}^{n_w}$, are unmeasured disturbances that are considered the $n_\theta = 6$ uncertain parameters:

$$\theta := [\theta_{res}^T, \theta_{top}^T]^T \quad (5.13)$$

The full set of parameter values and description can be seen in MATIAS *et al.* (2022), and the model implementation code is available on the "ProductionOptRig" Github page¹.

5.3.3 Optimization problem

To follow the same assumptions as MATIAS *et al.* (2022) and generate comparable results, the optimization objective function was selected as a weighted sum of the liquid flow rates:

$$\mathcal{J} = \mathbf{p}^T \mathbf{Q}_l \quad (5.14)$$

in which $\mathbf{p} \in \mathbb{R}^{n_w}$ are the liquid price vector. For illustration purposes, different liquid prices are considered for each well; in this study, the price vector was $\mathbf{p} = [20, 10, 30]^T$.

The constraints comprise the gas lift injection limits for each well and the gas availability constraint. The gas lift injection limits are defined as follows:

$$Q_g^{lb} \leq Q_{g,i} \leq Q_g^{ub}, \quad \forall i \in [1, 2, 3] \quad (5.15)$$

in which $Q_{g,i} := w_{g,i}/\rho_{g,i}$ is the gas flow rate for each well $i \in [1, 2, 3]$, Q_g^{lb} and Q_g^{ub} are the gas flow rate lower and upper bound, respectively equal to 1 slm and 5 slm. In addition, the gas availability constraint is defined as:

$$\sum_{i=1}^3 Q_{g,i} \leq Q_{max} \quad (5.16)$$

in which Q_{max} is the maximum gas throughput, which is considered 7.5 slm.

5.3.4 Hammerstein model identification

The Hammerstein model identification procedure was followed as described in Section 5.2.2. Independent disturbances were imposed on the gas injection flow

¹<https://github.com/Process-Optimization-and-Control/ProductionOptRig>

rate and the reservoir valve opening while keeping the pump pressure constant to generate the training data. The same input profile was applied to the three wells in the experimental rig; the input profile can be visualized in Figure 5.3.

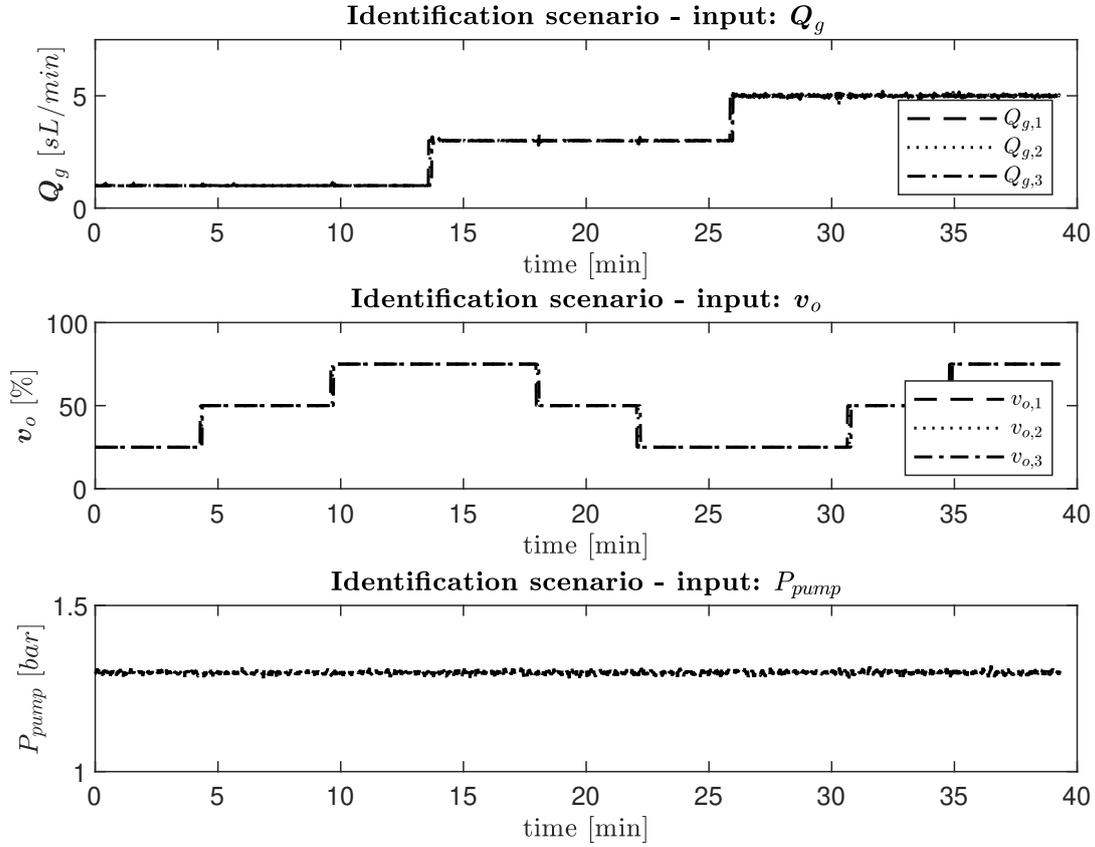


Figure 5.3: Identification scenario - input profile imposed to the experimental rig.

The pump pressure was kept at 1.3 bar , three values of gas injection flow rate 1.0 , 3.0 and 5.0 sl/min , and three values of reservoir valve opening 25 , 50 and 75% were applied. This single scenario explores nine different operational conditions, covering most of the feasible input range.

The measured variables can be visualized in Figure 5.4. It can be seen that the riser pressure is not much sensitive to the input variables. However, it can be noted that the noise level increases when the gas injection flow rate raises. This data set was used to solve the identification problem described in Equation 5.7, where the Hammerstein model is propagated over the whole training data horizon using the first point as the initial condition, and $W_{id} = I_6$.

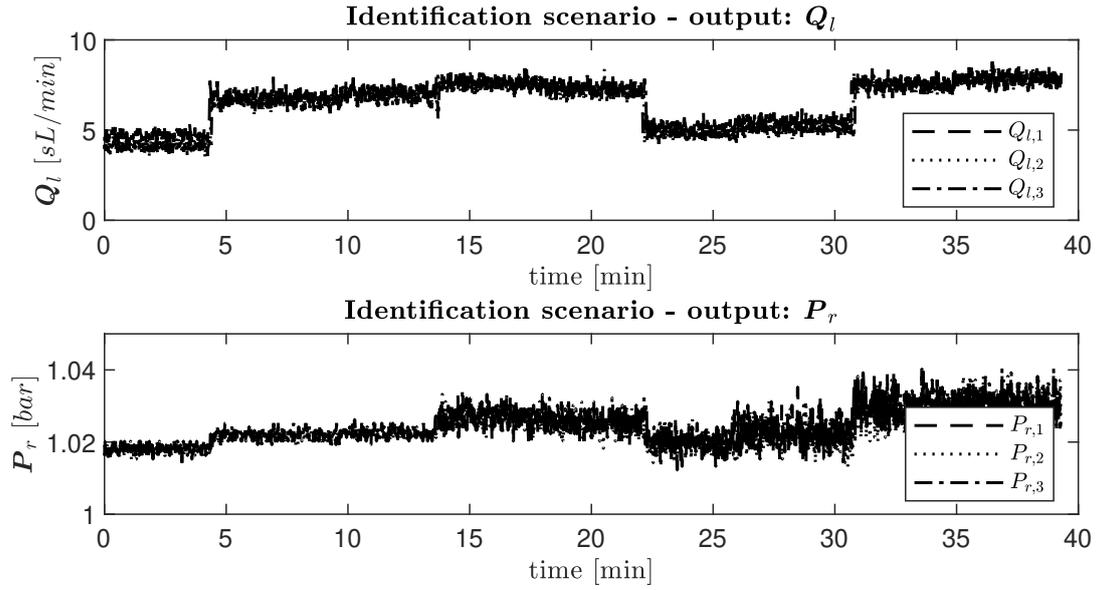


Figure 5.4: Identification scenario - output responses.

It can be seen that the dynamic responses of the plant are very fast, presenting stabilization periods of less than 1 *min*. This can be explained by the rig presenting small pipe holdups. Therefore, beyond the identified Hammerstein model, the steady-state model was tested as an instantaneous dynamic model using the same Hammerstein framework. The models tested are summarized below:

1. **Model 1:** Hammerstein model with the identified dynamic matrix A^h . The HROPA using this model is labeled as HROPA1 in the following sections;
2. **Model 2:** Hammerstein model with the dynamic matrix $A^h := 0$. This is essentially the steady-state model treated as a black-box model. The HROPA using this model is labeled as HROPA2 in the following sections.

The results related to the Hammerstein model identification are discussed in Section 5.4.1.

5.3.5 ROPA closed-loop disturbance scenario

The performance study was done under the same conditions as in MATIAS *et al.* (2022) as well as using the same execution period of 10 *s* for both HROPA approaches. Each experiment occurred in a 20-minute window, where the pump pressure P_{pump} was kept constant, and the reservoir valve opening profile changed according to Figure 5.5.

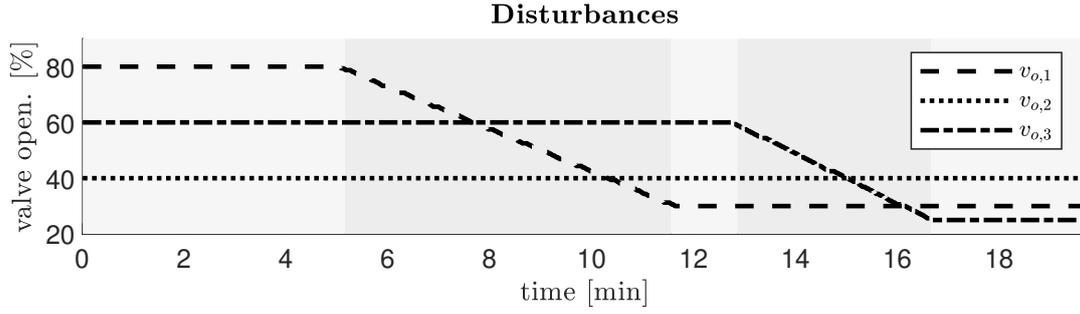


Figure 5.5: Reservoir valve opening profile. Different gray-shaded background highlights the five disturbance periods.

The scenario is divided into five periods, each period is highlighted in the figure by the use of different gray shades in the background. They are:

- From 0 to 5 min: valve openings kept constant at $v_o = [80, 60, 40]^T\%$;
- From 5 to 11 min: valve CV101 opening decreased linearly from 80 % to 30 %;
- From 11 to 13 min: valve openings kept constant at $v_o = [30, 60, 40]^T\%$;
- From 13 to 17 min: valve CV103 opening decreased linearly from 60 % to 25 %;
- From 17 to 20 min: valve openings kept constant at $v_o = [30, 60, 25]^T\%$;

As previously stated, the system responses to changes in gas flow rates are considerably fast. Therefore, the disturbance scenario is selected to emulate slow and persistent variation in the gas wells. This scenario can also be interpreted as well's depletion, which is the natural decrease of oil production over time that occurs in real gas-lifted systems.

The main goal of the experiment is to determine the economically optimal gas flow rate injection distribution among the wells during the disturbance scenario. To do so, the proposed HROPA approach is compared to the originally proposed ROPA presented in MATIAS *et al.* (2022). As HROPA differs from ROPA by the use of an approximate Hammerstein dynamic model in the EKF stage, the analysis carried out will consider the following items as criteria to demonstrate the HROPA concept:

- Ability to estimate the uncertain parameters accurately when compared to ROPA, which uses a rigorous dynamic model in the estimation step;
- Optimal input profiles similar to the ones computed by ROPA;
- Profitability difference between ROPA and HROPA.

It is expected that HROPA presents a lower economic performance than ROPA since a less accurate dynamic model is used. However, the lower effort required to develop the Hammerstein models compared to developing mechanistic dynamic models can compensate the lower economic performance in practical applications.

The results related to the HROPA in real-time and closed-loop with the experimental rig are discussed in Section 5.4.2.

5.3.6 EKF tuning

EKF tuning is one of the most critical factors for the success of ROPA/ HRTTO approaches. As discussed in MATIAS and LE ROUX (2018), this step must not be neglected since the EKF has a very important role in the algorithm stability. Failing to use proper tuning could hinder the operation and even completely destabilize the plant in closed-loop. However, to this date, the literature still lacks a systematic and generalized approach for EKF tuning, and this activity is often done by trial and error.

When a rigorous dynamic mechanistic model is available, the activity to find the first EKF tuning is done using the same model to emulate the plant in a simulated scenario. Once the first tuning is available, the ROPA algorithm can be coupled to the operation in an open loop for fine adjustments to the EKF tuning over the real operation and assessments of the optimal input profiles. After having confidence in the EKF tuning and optimization actions, the loop should be closed.

Considering that a rigorous dynamic mechanistic model is not available, the aforementioned procedure can still be followed but using the approximated dynamic model to emulate the plant. Therefore, special care should be given to the fine adjustments directly in the plant while in open loop. In this work, however, there was a big time gap between the experiments of the ROPA and the HROPA algorithm. By the time the HROPA algorithm was tested, the experimental rig had a lower availability for continuous operation, which limited the possibility of tuning adjustments. Therefore, the tuning procedure was only performed in a simulated environment by using the Hammerstein dynamic model to emulate the plant behavior. No fine-tuning adjustments were applied directly to the experimental rig.

The Hammerstein EKF tuning obtained in a simulated way was:

$$\mathbf{Q}_x = \text{diag} ([3e^{-1}, 3e^{-1}, 3e^{-1}, 7e^{-6}, 7e^{-6}, 7e^{-6}]^T) \quad (5.17a)$$

$$\mathbf{Q}_\theta = \text{diag} ([1e^1, 1e^1, 1e^1, 1e^0, 1e^0, 1e^0]^T) \quad (5.17b)$$

$$\mathbf{R} = \text{diag} ([3e^{-1}, 3e^{-1}, 3e^{-1}, 7e^{-6}, 7e^{-6}, 7e^{-6}]^T) \quad (5.17c)$$

(5.17d)

in which $\mathbf{Q} = \text{diag}([\mathbf{Q}_x, \mathbf{Q}_\theta]^T)$ is the augmented state noise covariance matrix and \mathbf{R} is the measurement noise covariance matrix. We refer to DELOU *et al.* (2021c) for the complete Hammerstein EKF formulation.

5.4 Experimental results

5.4.1 Identification results

The identification results of the two Hammerstein models for the output variables are presented in Figures 5.6 and 5.7. These output profiles are obtained by applying the input sequences shown in Figure 5.3 to the experimental rig.

From Figure 5.6, it is possible to see that both models presented a good fit for the liquid flow rates in all three wells.

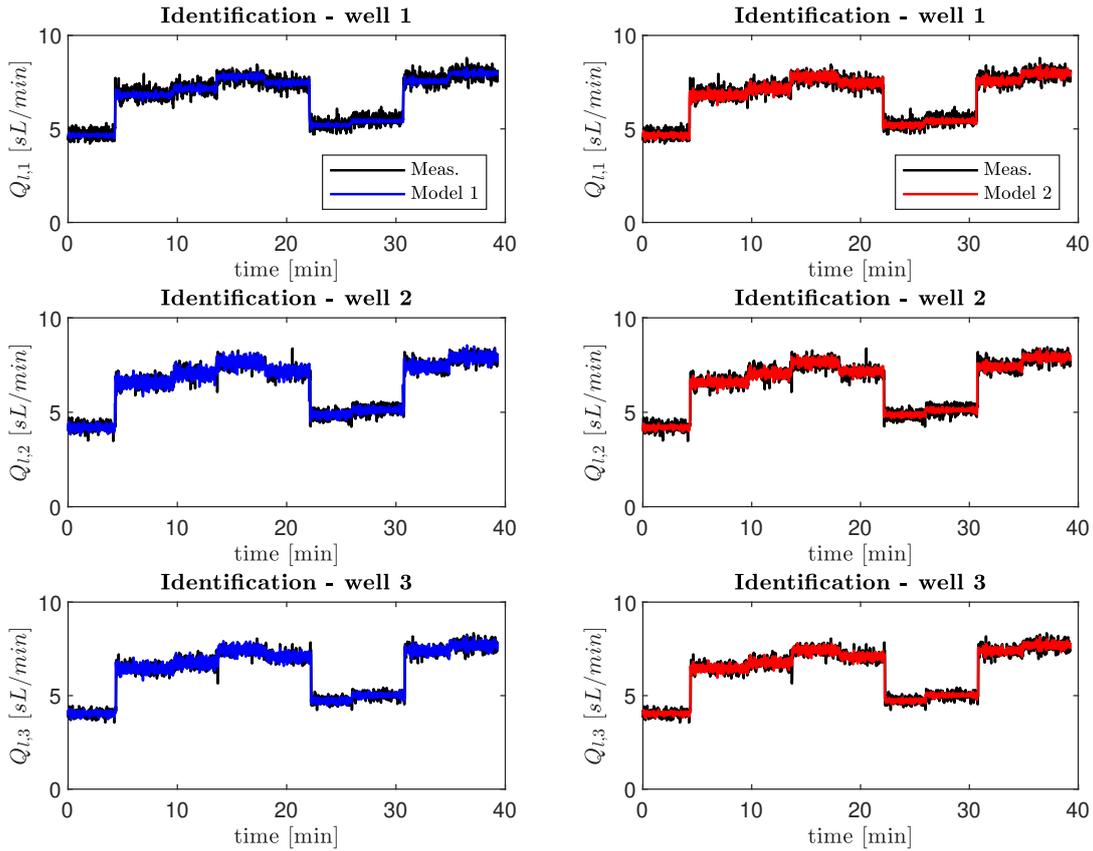


Figure 5.6: Identification result - liquid flow rate. Blue line denotes the response from Model 1, which contains the identified matrix \mathbf{A}^h , and red line denotes the response from Model 2, which considers only the steady-state model.

The results for the riser pressure are presented in Figure 5.7. Both models presented a fair performance in predicting the riser pressure (note that the y-axis

range in the figure is only 0.05 bar), but model 1 response was noisier than the measurement. In contrast, model 2 response was slightly less noisy than the measurement.

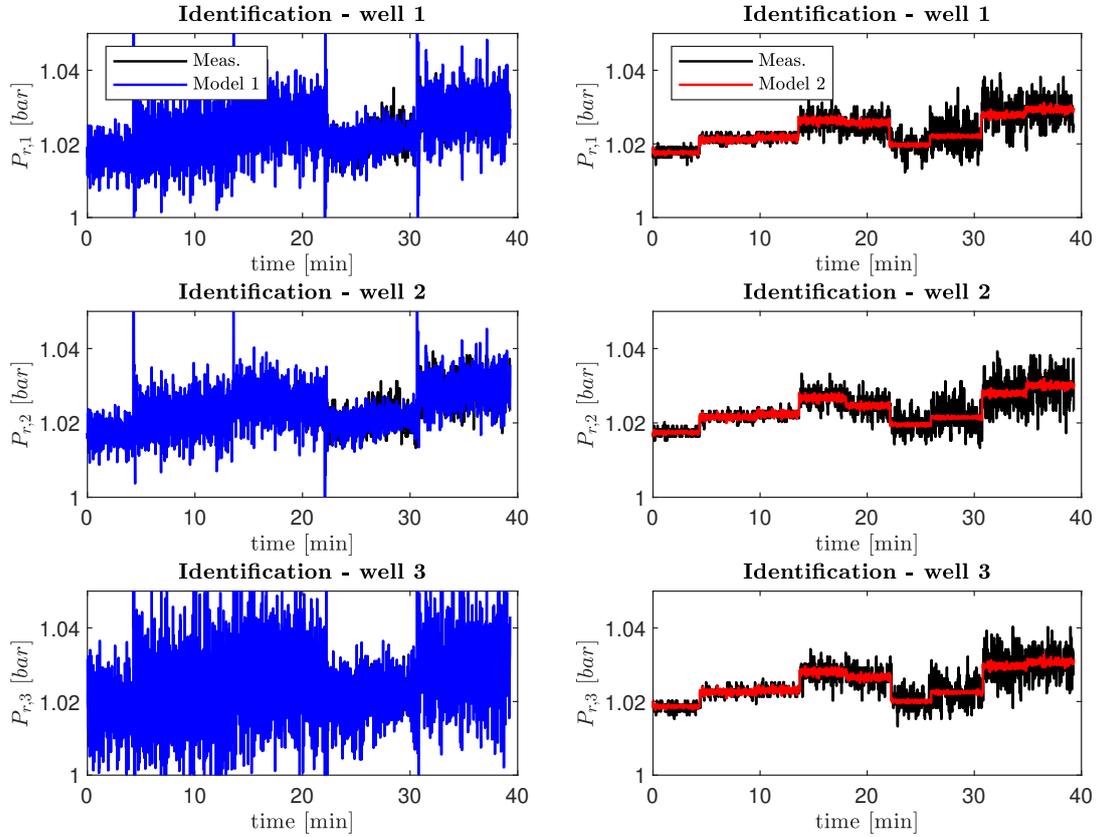


Figure 5.7: Identification result - riser pressure. Blue line denotes the response from Model 1, which contains the identified matrix A^h , and red line denotes the response from Model 2, which considers only the steady-state model.

It is important to note that the identification optimization problem is not convex; hence, there is a possibility that multiple models would generate a similar performance. This aspect was not explored here as the goal was to have a reasonable model that could approximate the plant dynamics rather than having the best possible model. However, in real applications, one should consider exploring using global optimizers and different weights on the objective function.

5.4.2 ROPA and discussion

The performance of the proposed HROPA using the approximate Hammerstein dynamic model in the parameter estimation stage is compared to the ROPA algorithm using a first-principles dynamic model implemented in MATIAS *et al.* (2022). As described in Section 5.3.5, the same disturbance scenario and the same execution period of 10 s were used to keep some level of comparability. However, it is important to note that the result presented here as ROPA is the one presented

by MATIAS *et al.* (2022), the experiment was not rerun for this chapter. Also, it is important to mention that the time difference between the HROPA and ROPA experiments was greater than one year, which is a considerably high amount of time. For this reason, it is possible that some aspects of the experimental rig may have changed, such as valve and sensor calibration and environmental conditions. For example, the gas flow meter devices associated with the controllers FIC 104, 105, and 106 are very sensitive to moisture. If moisture accumulates in the air injection controllers, the flow rate measurements drift from the true values. It may take a considerable amount of time until this improper behavior is identified and the sensors are re-calibrated.

Additionally, it is expected that the ROPA algorithm performs better than HROPA for two reasons. The first is that it uses a better dynamic model, while HROPA uses a fair approximation. The second is that before the ROPA runs, there were some rounds of EKF fine-tuning directly on the rig, which has not happened for the HROPA runs. The former ran with tuning derived in a simulated environment. Therefore, for the aforementioned reasons, the results will be compared more in terms of general trends than quantitative accuracy.

Figures 5.8 and 5.9 present the parameter estimation results for ROPA, HROPA1, and HROPA2. These results are particularly important since an accurate parameter estimation will define an accurate model adaptation, and consequently, an accurate optimal solution can be found.

The estimation of the reservoir valve coefficients is presented in Figure 5.8. For all wells, the HROPA1 and HROPA2 results presented a fairly similar result to the ROPA result, following the same trends with a similar chattering level. Both HROPA1 and HROPA2 presented a responsive behavior to the ramp disturbances imposed on wells 1 and 3 (see Figure 5.5). For both disturbances, the profiles of the reservoir valve coefficient estimates follow the openings.

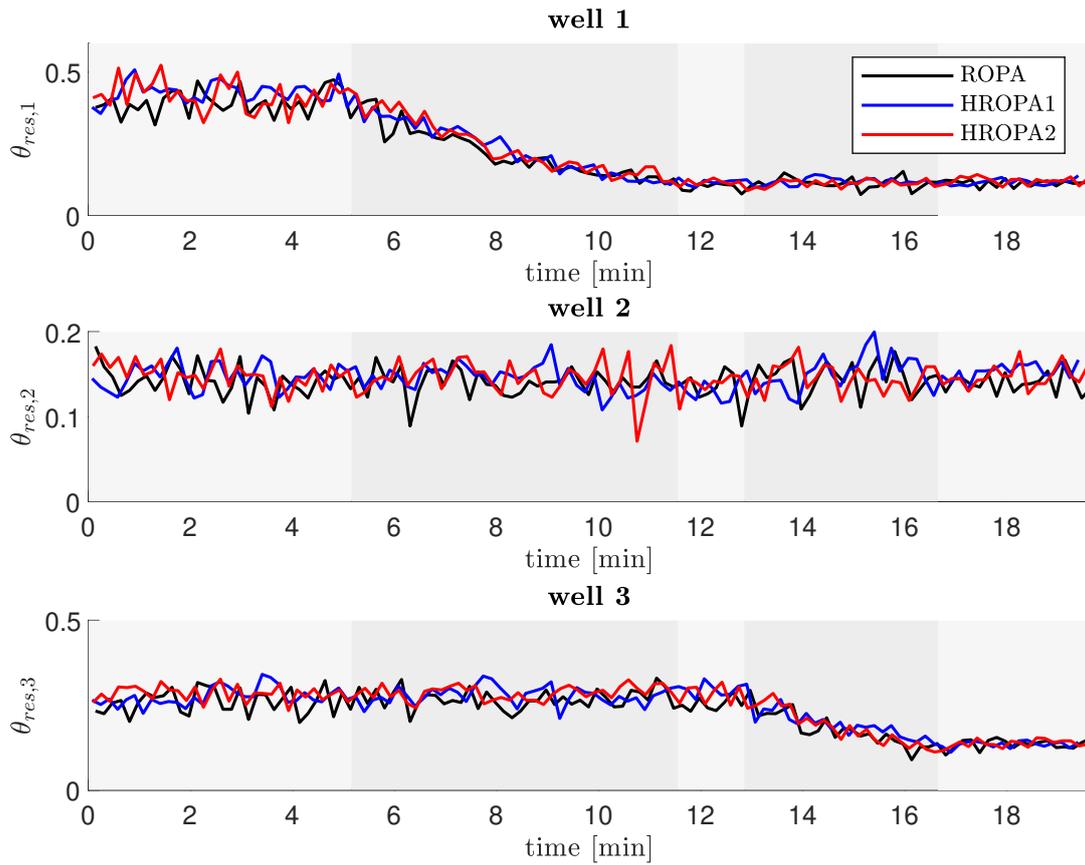


Figure 5.8: Parameter estimation - reservoir valve coefficients.

The estimation of the riser valve coefficients estimation is presented in Figure 5.9. Although these coefficients should not change much under the imposed disturbance scenarios, HROPA1 and HROPA2 presented an excessively noisy estimation, noting that HROPA1 was even noisier than HROPA2. This behavior was caused by poor EKF tuning since a fine adjustment was not made directly in the experimental rig operation, contrary to ROPA, which had a fine-tuning step. It is clear that, on average, the correct parameter estimation value is being found, but this excessive amount of noise can affect the overall economic performance of the algorithm.

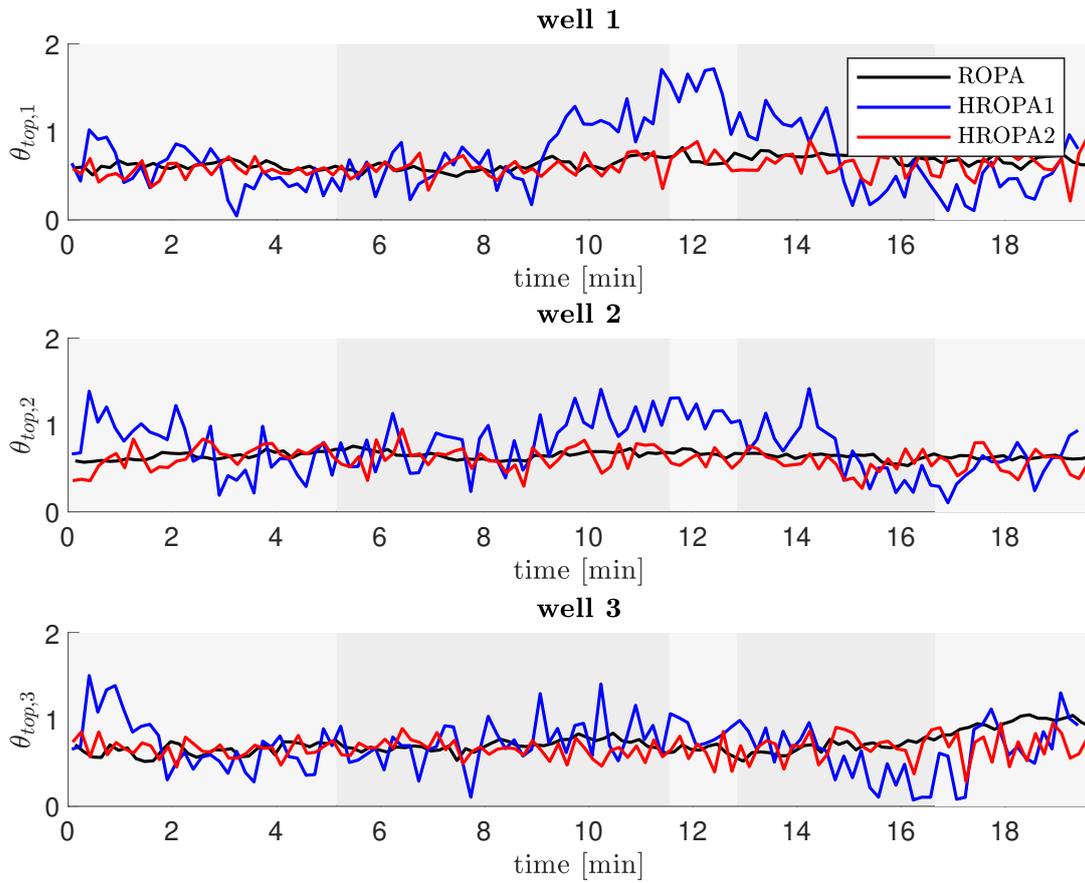


Figure 5.9: Parameter estimation - top valve coefficients.

The gas injection flow rate optimal input profile is presented in Figure 5.10. The optimal profiles of HROPA1 and HROPA2 presented a similar trend compared to ROPA. It is possible to see the noisy estimation of the riser valve coefficients has impacted the HROPA solutions during some specific periods. However, the overall changes in the optimal gas flow distribution were successfully captured under the disturbance scenarios affecting the system.

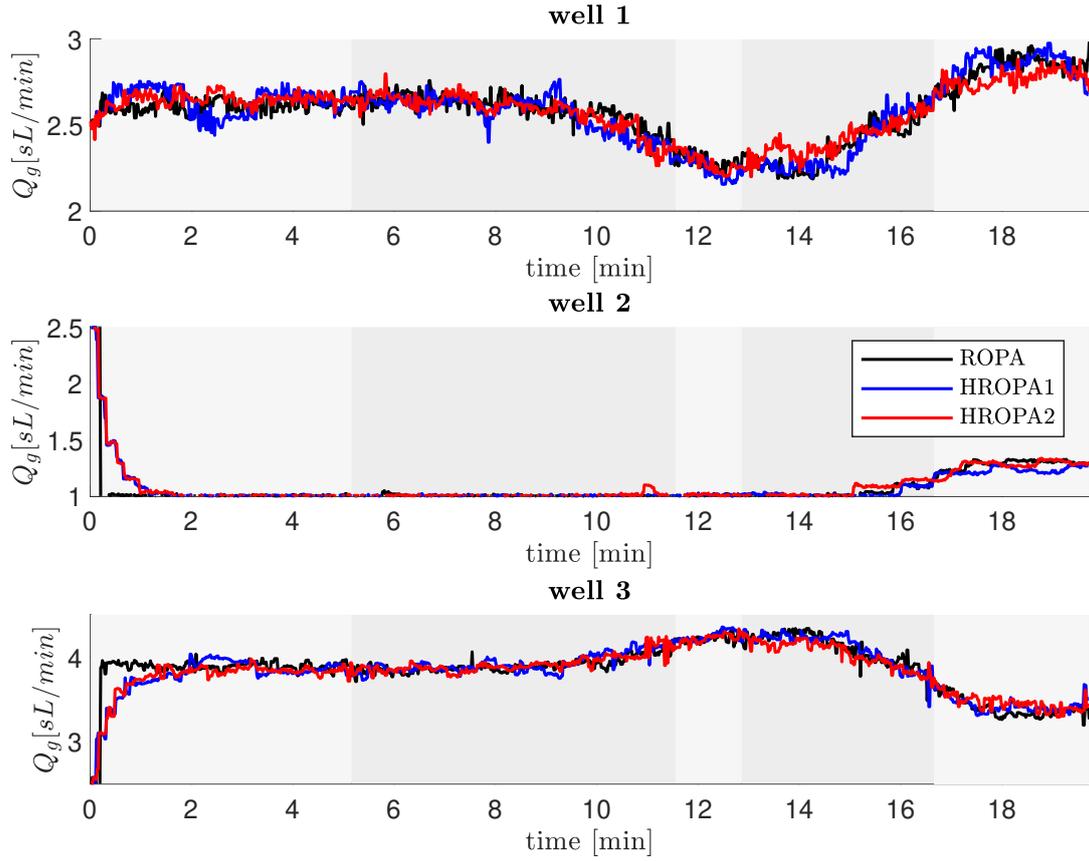


Figure 5.10: Optimal input profile - gas injection flow rate.

The economic performance of the HROPA algorithms was assessed and presented in Figure 5.11. The figure presents the absolute profit function defined by Equation 5.14 and the percentage profit difference to ROPA, $J_{(\cdot)}^{diff}$, defined in Equation 5.18. A moving average with a 30s time-frame window was implemented to improve the visualization of the percentage profit difference.

$$J_j^{diff} = \frac{|J_j - J_{ROPA}|}{J_{ROPA}}, \forall j \in [ROPA, HROPA1, HROPA2] \quad (5.18)$$

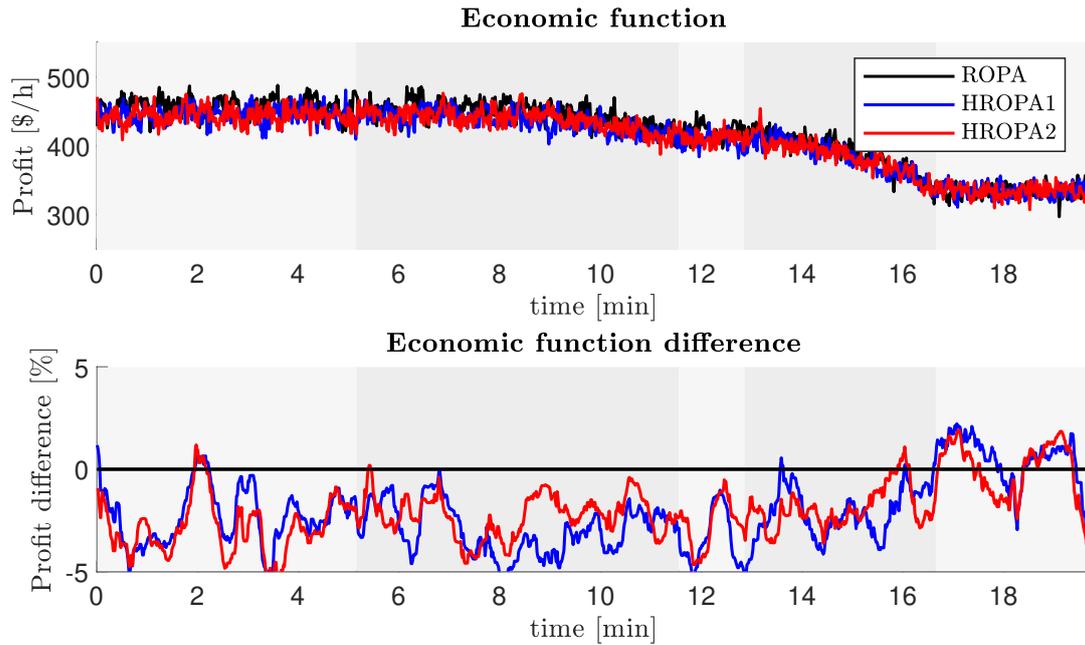


Figure 5.11: Profit and average profit difference comparison between approaches.

Although the profit functions are very similar for all approaches, it is possible to see that ROPA yields a slightly larger profit value, as expected. From the average profit difference, we can see that HROPA1 and HROPA2 presented an overall poorer economic performance; the average difference was -1.98% for HROPA1 and -1.85% for HROPA2. These differences can be interpreted as the potential operating cost loss associated with using an approximate dynamic instead of developing a mechanistic dynamic model. On the other hand, obtaining the approximate model requires considerably less investment in terms of man-hours and it is much easier to maintain. This trade-off should be addressed during the RTO project phase and involve both technical and management aspects that should be taken into account.

However, considering the current experimental setup, other factors also contributed to those differences, such as the identified Hammerstein model, which presented a high noise for the riser pressure, and the Hammerstein EKF tuning, which was only derived in a simulated environment with no fine-tuning stage directly on the experimental rig. Therefore, it is safe to say that these differences could be overestimated due to the limitations of the current HROPA experiments and the time difference from the ROPA experiment.

Finally, it is important to highlight that HROPA2, which is a ROPA framework using the static model directly in an EKF algorithm, presented a surprisingly good performance considering that there was no effort to derive a dynamic mechanistic model or to identify an approximate Hammerstein model. This was valid since the current system presents fast dynamics due to small material holdup inside

the pipelines, and the assumption of using the static model as an instantaneous dynamic model is fairly acceptable. These results indicate a successful application of EKF using a static model, even in a drifting disturbance scenario. Therefore, for fast dynamic modes, a classic RTO framework may be turned into a ROPA by implementing the static model in an EKF algorithm and removing the steady-state detection stage. Another benefit of this change is that the EKF algorithm runs much faster than the classic static data reconciliation and parameter estimation algorithms commonly employed in classic RTO frameworks, e.g. least-squares estimation.

Although the computational cost of the algorithm was not evaluated during the closed-loop experiments, this evaluation was done in a simulated environment. Table 5.1 presents the average, minimum, and maximum computational cost in s of the EKF and optimization stages for ROPA and HROPA.

Table 5.1: Computational cost comparison between ROPA and HROPA

Stage	Average time (s)	Minimum time (s)	Maximum time (s)
ROPA - EKF	0.051	0.041	0.125
HROPA - EKF	0.014	0.011	0.034
ROPA - Optimization	0.040	0.037	0.081
HROPA - Optimization	0.042	0.038	0.100

Given the execution period of 10 s , it is safe to say that both ROPA and HROPA presented a computational cost much lower than the real-time closed-loop requirement. Both approaches present a similar computational cost in terms of the optimization step, which is expected since they execute the same optimization problem. However, we can see that the Hammerstein EKF is 3.6 times faster than the rigorous model EKF. This can be explained by two reasons; the first is that the Hammerstein model is a reduced-order model compared to the rigorous one, presenting a lower number of state variables. The second reason is that the Hammerstein EKF required fewer sensitivity evaluations compared to the rigorous model, the only sensitivity evaluation needed is in reference to the outputs and the parameters, while the rigorous model EKF requires the computation of the states Jacobian matrix.

5.5 Conclusion

This chapter presents an implementation of Real-time Optimization with Persistent Parameter Adaptation (ROPA) using transient data in the absence of a dy-

dynamic mechanistic model on a small-scale pilot plant. An approximate dynamic Hammerstein structure that takes advantage of the available static model replaces the dynamic mechanistic model in the extended Kalman filter (EKF) module. This is possible since the Hammerstein structure is compatible with the static model used for optimization; it preserves the parameters observability from the static model; and presents fairly accurate dynamics compared to the real plant.

The main contribution of this chapter is proving the concept of Hammerstein ROPA (HROPA) in a real system. The methodology presents increased potential applicability to large-scale systems with significantly fewer man-hours effort than the original ROPA methodology because it does not require a dynamic mechanistic model. Two HROPA are compared, one using a virtual dynamic matrix identified using plant data (HROPA1) and another using the static model directly as it was an instantaneous dynamic model (HROPA2). The results showed that both HROPA approaches presented slightly lower economic performance than ROPA but a similar ability to estimate uncertain parameters and obtain similar optimal input profiles. Nonetheless, the dynamic model approximation is not the only factor that explains the lower economic performance since the EKF tuning and the experimental conditions also impacted the comparison. Therefore, it is fair to say that HROPA can produce a similar performance to ROPA with much less effort in the dynamic model design stage.

Another contribution was to show that the static model can be directly used in the EKF algorithms when the system presents fairly fast dynamics. For this kind of system, there would be no effort to develop a mechanistic dynamic model, and HROPA could be directly applied. For these reasons, we believe that our research improves the applicability of ROPA for systems where developing a dynamic model is not economically feasible and establishes the basis for future applications of HROPA in large-scale systems.

Chapter 6

Output Modifier adaptation based on Gaussian Process: Simultaneous use in Real-Time Optimization and Hammerstein NMPC

A version of this chapter was presented at the 20th European Control Conference (ECC 22).

6.1 Introduction

This chapter further advances the use of GPs inside MA schemes. We propose a GP-based Output Modifier adaptation (MAy-GP) and compare it to the previously proposed GP-based Modifier adaptation (MA-GP). The proposed methodology also considers trust-region concepts, which increases the algorithm's robustness and presents an inherent exploration capability. Also, we expanded the control methodology proposed by DELOU *et al.* (2021c) by incorporating a way to deal with model structural uncertainty in the proposed Hammerstein NMPC by including the correction term from MAy-GP. The benefit of using MAy-GP over MA-GP comes from the fact that MAy-GP corrects the model output variables instead of correcting the objective function and constraints of the economic optimization problem. That is, MAy-GP provides a more generic correction capability since it acts directly on the system model in contrast with the MA-GP which provides correction terms specifically for each optimization problem of interest. Therefore, these model correction terms can be interchangeable with other model-based techniques, such as process control and observers.

The remainder of the chapter is organized as follows: Section 6.2 provides a

brief introduction to the MA technique and on GPs; Section 6.3 describes the proposed methodology and the contributions of this chapter; Section 6.4 presents the results of the application of the proposed methodology to the Williams-Otto reactor; and Section 6.4 is the conclusion.

6.2 Preliminaries

6.2.1 Problem Formulation

The plant optimization problem consists of minimizing an economic performance index in the presence of constraints:

$$\min_{\mathbf{u} \in \mathbb{U}} \mathcal{J}(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \quad (6.1a)$$

$$\text{s.t. } g_i(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \leq 0, \forall i \in \mathbb{I}_{n_g}^+ \quad (6.1b)$$

in which, $\mathbf{u} \in \mathbb{U}$ are the input variables, \mathbb{U} is the input free moving space delimited by lower and upper bounds $\mathbb{U} = \{\mathbf{u} \in \mathbb{R}^{n_u} : \mathbf{u}^{lb} \leq \mathbf{u} \leq \mathbf{u}^{ub}\}$; $\mathbf{y}_p \in \mathbb{R}^{n_y}$ are the measured output variables, $\mathcal{J} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ is the cost function and $g_i : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ are the $i \in \mathbb{I}_{n_g}^+$ constraint functions, such that $\mathbb{I}_{n_g}^+ = \{i : 1, \dots, n_g\}$.

In practice, however, Problem 6.1a is unknown since a precise steady-state input-output map, $\mathbf{y}_p(\mathbf{u})$ is not available. In fact, we generally have only an approximate steady-state model that can be represented by the following nonlinear algebraic equation system in the state-space form:

$$\mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}) \quad (6.2a)$$

$$\mathbf{y} = \mathcal{G}(\mathbf{x}, \mathbf{u}) \quad (6.2b)$$

in which, $\mathbf{x} \in \mathbb{X}$ are the state variables, such that $\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{x}^{lb} \leq \mathbf{x} \leq \mathbf{x}^{ub}\}$; $\mathbf{y} \in \mathbb{R}^{n_y}$ are the predicted output variables, $\mathcal{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ represent the system of algebraic equations and $\mathcal{G} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$ represent the output algebraic equation system. Therefore, the model-based optimization problem, considering the notation introduced by Equation 6.2, is given by:

$$\min_{\mathbf{u} \in \mathbb{U}, \mathbf{x} \in \mathbb{X}} J(\mathbf{u}, \mathbf{y}) \quad (6.3a)$$

$$\text{s.t. } g_i(\mathbf{u}, \mathbf{y}) \leq 0, \forall i \in \mathbb{I}_{n_g}^+ \quad (6.3b)$$

$$\mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}) \quad (6.3c)$$

$$\mathbf{y} = \mathcal{G}(\mathbf{x}, \mathbf{u}) \quad (6.3d)$$

Frequently, the optimal solutions of Problems 6.1a and 6.3 do not match due to model parameter and/or structural uncertainties, disturbances, and process noise (MARCHETTI *et al.*, 2016). Therefore, the main goal of RTO is to use online process measurements to continuously adapt Problem 6.3 in order to reach plant optimality of Problem 6.1a.

6.2.2 Modifier adaptation

In Modifier adaptation techniques, modifiers are included in the model-based optimization problem in order to achieve plant necessary conditions of optimality (NCO) upon convergence. These modifiers can play the role of direct corrections to the cost and constraint functions, such that:

$$\mathbf{u}_{k+1}^* = \arg \min_{\mathbf{u} \in \mathbb{U}, \mathbf{x} \in \mathbb{X}} [J + \mu_k^J](\mathbf{u}, \mathbf{y}) \quad (6.4a)$$

$$\text{s.t. } [g_i + \mu_k^{g_i}](\mathbf{u}, \mathbf{y}) \leq 0, \forall i \in \mathbb{I}_{n_g}^+ \quad (6.4b)$$

$$\mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}) \quad (6.4c)$$

$$\mathbf{y} = \mathcal{G}(\mathbf{x}, \mathbf{u}) \quad (6.4d)$$

or indirect corrections applied to the output variables in a way that the modified outputs affect the cost and constraint functions, that is:

$$\mathbf{u}_{k+1}^* = \arg \min_{\mathbf{u} \in \mathbb{U}, \mathbf{x} \in \mathbb{X}} J(\mathbf{u}, \mathbf{y}_m) \quad (6.5a)$$

$$\text{s.t. } g_i(\mathbf{u}, \mathbf{y}_m) \leq 0, \forall i \in \mathbb{I}_{n_g}^+ \quad (6.5b)$$

$$\mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}) \quad (6.5c)$$

$$\mathbf{y}_m = \mathcal{G}(\mathbf{x}, \mathbf{u}) + \mu_k^y(\mathbf{u}) \quad (6.5d)$$

For a generic function $f(\cdot)$, μ_k^f is defined as zeroth- and first-order input-affine corrections, such that:

$$\mu_k^f := f_p(\mathbf{u}_k) - f(\mathbf{u}_k) + [\nabla_{\mathbf{u}} f_p(\mathbf{u}_k) - \nabla_{\mathbf{u}} f(\mathbf{u}_k)]^T \mathbf{u} \quad (6.6)$$

We will refer to the first strategy, described by Problem 6.4, just by the abbreviation MA, and to the Output Modifier adaptation by MAy. It can be shown that, under certain hypotheses and accurate gradient estimation, both MA and MAy are capable of matching the plant NCO (MARCHETTI *et al.*, 2016). However, the main drawback of MA and MAy is the gradient estimation from data at every instant k .

6.2.3 Gaussian Process Approximation

Given an unknown function $f : \mathbb{R}^{n_u} \rightarrow \mathbb{R}$ and $y = f(\mathbf{u}) + v$, in which v is a zero-mean white noise with unknown variance σ^2 , we consider an available N -sized input-output data set generated by $f(\cdot)$, $\mathbf{U} := [\mathbf{u}_1, \dots, \mathbf{u}_N] \in \mathbb{R}^{n_u, N}$ and $\mathbf{y} := [y_1, \dots, y_N]^T \in \mathbb{R}^N$. Gaussian process regression is a kernel-based method that uses all available data to establish a relationship between \mathbf{U} and \mathbf{y} in order to approximate the unknown input-output map $f(\cdot)$. Due to its stochastic characteristics, GP differs from most parametric and non-parametric deterministic approaches in the sense that, at an arbitrary input \mathbf{u} , it not only gives an output estimate but also its uncertainty:

$$f(\mathbf{u})|\mathbf{U}, \mathbf{y} \sim \mathcal{GP}(\mu_f(\mathbf{u}), \sigma_f^2(\mathbf{u})) \quad (6.7)$$

in which, μ_f is the posterior mean function and σ_f^2 is the posterior variance function computed by:

$$\mu_f(\mathbf{u}) = \mathbf{r}(\mathbf{u}, \mathbf{U})\mathbf{K}(\mathbf{U})^{-1}\mathbf{y} + c \quad (6.8)$$

$$\sigma_f^2(\mathbf{u}) = \sigma_n^2(\mathbf{u}) - \mathbf{r}(\mathbf{u}, \mathbf{U})\mathbf{K}(\mathbf{U})^{-1}\mathbf{r}(\mathbf{u}, \mathbf{U})^T \quad (6.9)$$

in which, $K_{i,j} := k(\mathbf{u}_i, \mathbf{u}_j) + \sigma_v^2 \delta_{i,j}$, $\forall (i, j) \in 1, \dots, N^2$, $\delta_{i,j}$ is the Kronecker's delta function, c is a constant mean function, $\mathbf{r}(\mathbf{u}, \mathbf{U}) := [k(\mathbf{u}, \mathbf{u}_1), \dots, k(\mathbf{u}, \mathbf{u}_N)]$ and $k(\cdot, \cdot)$ is the squared-exponential kernel function:

$$k(\mathbf{u}, \bar{\mathbf{u}}) := \sigma_n^2 \exp\left(-\frac{1}{2}(\mathbf{u} - \bar{\mathbf{u}})^T \mathbf{\Lambda}(\mathbf{u} - \bar{\mathbf{u}})\right) \quad (6.10)$$

in which, σ_n^2 is the covariance and $\mathbf{\Lambda} := \text{diag}(\lambda_1, \dots, \lambda_{n_u})$ is a scaling matrix. The GP's hyperparameters $\Psi := [c, \sigma_n, \sigma_v, \lambda_1, \dots, \lambda_{n_u}]^T$ are usually estimated by a maximum likelihood approach, in which the log-likelihood of the data is:

$$\mathcal{L} := -\frac{1}{2} \ln(|\mathbf{K}(\mathbf{U})|) - \frac{1}{2}(\mathbf{y} - \mathbf{1}c)^T \mathbf{K}(\mathbf{U})^{-1}(\mathbf{y} - \mathbf{1}c) \quad (6.11)$$

In the context of RTO, the unknown function to be estimated can be the plant steady-state map, some or all of the optimization constraints, the objective function, or the mismatch of these functions and an available model (DE AVILA FERREIRA *et al.*, 2018). DEL RIO CHANONA *et al.* (2021) showed that there is enhanced reliability in mapping the plant-model mismatch instead of a total model-

free strategy. We refer to RASMUSSEN and WILLIAMS (2005) for a more in-depth exposition of GP and its broad range of possibilities in the field of Machine Learning.

6.3 Methodology

6.3.1 Trust-region MA-GP Scheme

The first to propose the use of GP in the context of MA was DE AVILA FERREIRA *et al.* (2018), here called MA-GP. Later, DEL RIO CHANONA *et al.* (2019) expanded the frameworks by introducing the trust-region approach and how to include GP uncertainty directly into the optimization problem. DEL RIO CHANONA *et al.* (2021) expanded the methodology in theoretical terms, including the concept of Acquisition Functions to promote exploration characteristics to the framework. The main idea is to use GPs to model the mismatch between the cost and constraint functions separately, such that:

$$J_p - J \sim \mathcal{GP}(\mu^J, (\sigma^J)^2) \quad (6.12a)$$

$$g_{p,i} - g_i \sim \mathcal{GP}(\mu^{g_i}, (\sigma^{g_i})^2), \forall i \in \mathbb{I}_{n_g}^+ \quad (6.12b)$$

in which, for a generic function $f(\cdot)$ and in the k^{th} iteration of the RTO, μ_k^f and σ_k^f represent the mean and the standard deviation of the GP considering the data set $(\mathbf{U}_k, f_p(\mathbf{u}_k) - f(\mathbf{u}_k))$. The trust-region MA-GP problem considering the GP mean as a higher-order correction term is:

$$\Delta \mathbf{u}_{k+1}^* = \arg \min_{\Delta \mathbf{u}, \mathbf{x} \in \mathbb{X}} \tilde{J} := [J + \mu_k^J](\mathbf{u}_k + \Delta \mathbf{u}, \mathbf{y}) \quad (6.13a)$$

$$\begin{aligned} \text{s.t. } \tilde{g}_i &:= [g_i + \mu_k^{g_i}](\mathbf{u}_k + \Delta \mathbf{u}, \mathbf{y}) \leq 0, \\ &\forall i \in \mathbb{I}_{n_g}^+ \end{aligned} \quad (6.13b)$$

$$\mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}_k + \Delta \mathbf{u}) \quad (6.13c)$$

$$\mathbf{y} = \mathcal{G}(\mathbf{x}, \mathbf{u}_k + \Delta \mathbf{u}) \quad (6.13d)$$

$$\|\Delta \mathbf{u}\| \leq \Delta_k, \quad (6.13e)$$

$$\mathbf{u}_k + \Delta \mathbf{u} \in \mathbb{U} \quad (6.13f)$$

in which, $\Delta \mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of input incremental from the last implemented inputs which is a decision variable in the trust-region scheme, $\Delta_k \geq 0$ is the trust-region radius in RTO iteration k and the procedures of updating the training data set and the trust-region follows the Algorithm 1 described by DEL RIO CHANONA *et al.* (2021).

6.3.2 Proposed Trust-region MAy-GP Scheme

In the present chapter, we proposed to include the GP correction in the indirect manner of MAy. That is, modeling the mismatch between the output measurements and the model prediction, such that:

$$y_{p,i} - y_i \sim \mathcal{GP}(\mu^{y_i}, (\sigma^{y_i})^2), \forall i \in \mathbb{I}_{n_y}^+ \quad (6.14)$$

Therefore, the trust-region MAy-GP problem can be written as:

$$\Delta \mathbf{u}_{k+1}^* = \arg \min_{\Delta \mathbf{u}, \mathbf{x} \in \mathbb{X}} \tilde{J} := J(\mathbf{u}_k + \Delta \mathbf{u}, \tilde{\mathbf{y}}) \quad (6.15a)$$

$$\begin{aligned} \text{s.t. } \tilde{g}_i &:= g_i(\mathbf{u}_k + \Delta \mathbf{u}, \tilde{\mathbf{y}}) \leq 0, \\ &\forall i \in \mathbb{I}_{n_g}^+ \end{aligned} \quad (6.15b)$$

$$\mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}_k + \Delta \mathbf{u}) \quad (6.15c)$$

$$\tilde{\mathbf{y}} = \mathcal{G}(\mathbf{x}, \mathbf{u}_k + \Delta \mathbf{u}) + \boldsymbol{\mu}_k^y(\mathbf{u}) \quad (6.15d)$$

$$\|\Delta \mathbf{u}\| \leq \Delta_k, \quad (6.15e)$$

$$\mathbf{u}_k + \Delta \mathbf{u} \in \mathbb{U} \quad (6.15f)$$

in which, $\tilde{\mathbf{y}} \in \mathbb{R}^{n_y}$ are the modified output variables and $\boldsymbol{\mu}_k^y(\mathbf{u}) := [\mu_k^{y_1}(\mathbf{u}), \dots, \mu_k^{y_{n_y}}(\mathbf{u})]^T$.

The main motivation for doing MAy instead of MA is the flexibility that comes with the use of the output modifiers. This adaptation can be used not only for optimization purposes but also for any model-based techniques, such as controllers and observers. This will be shown by the formulation of an NMPC with output modifiers.

6.3.3 Proposed NMPC with GP output modifiers formulation

In the context of HRTO, DELOU *et al.* (2021c) proposed the use of a Hammerstein structure to approximate the process dynamic model and enable the use of dynamic observers, such as the EKF, without the need to develop a rigorous dynamic process model based on first-principles. The Hammerstein structure takes advantage of the available static model and combines it with a linear auto-regressive exogenous (ARX) model to provide approximate dynamics that have to be identified from plant data. The Hammerstein structure is:

$$\mathbf{y}_{k+1}^h = \mathbf{A}^h \mathbf{y}_k^h + \mathbf{B}^h \mathbf{y}^s(\mathbf{u}_k) \quad (6.16)$$

in which, $\mathbf{y}_{k+1}^h \in \mathbb{R}^{n_y}$ is the set of output variables predicted by the Hammerstein

model at instant k , $\mathbf{y}^s(\mathbf{u}_k)$ is the respective steady-state of the output variables calculated by the available static model represented by Equation 6.2 given by the inputs at instant k . The dynamic matrices \mathbf{A}^h and $\mathbf{B}^h := \mathbf{I}_{n_y} - \mathbf{A}^h$ have dimension $n_y \times n_y$ and are identified from plant data around a single operating point.

Here, we propose to extend this structure by including the GP correction term, allowing the Hammerstein model to be applied even in a scenario of structural uncertainty:

$$\mathbf{y}_{k+1}^h = \mathbf{A}^h \mathbf{y}_k^h + \mathbf{B}^h [\mathbf{y}^s + \boldsymbol{\mu}^y](\mathbf{u}_k) \quad (6.17)$$

Therefore, the proposed structure represented by Equation 6.17 can be used for any model-based techniques that require a dynamic model, such as predictive controllers and observers. Even for methods that require successive linearization, this model is still handy since the static model can be linearized by an Automatic Differentiation framework and the GP presents an analytical derivative.

The following optimization problem represents the proposed infinite-horizon NMPC, which uses the nonlinear Hammerstein structure modified by the MAy-GP regressor as a predictive model and the trust region provided by the RTO.

$$\begin{aligned} \min_{\Delta \mathbf{u}} \quad & \sum_{j=1}^{N-1} (\mathbf{y}_{k+j} - \mathbf{y}_k^{sp})^T \mathbf{W}_y (\mathbf{y}_{k+j} - \mathbf{y}_k^{sp}) + \\ & \sum_{j=0}^{N-1} (\mathbf{u}_{k+j} - \mathbf{u}_k^{tg})^T \mathbf{W}_u (\mathbf{u}_{k+j} - \mathbf{u}_k^{tg}) + \\ & \sum_{j=0}^{N-1} \Delta \mathbf{u}_{k+j}^T \mathbf{W}_{\Delta u} \Delta \mathbf{u}_{k+j} + \mathbf{y}_{k+N}^T \mathbf{P} \mathbf{y}_{k+N} \end{aligned} \quad (6.18a)$$

$$\text{s.t. } \forall j \in [1, 2, \dots, N-1]$$

$$\mathbf{y}_{k+j}^h = \mathbf{A}^h \mathbf{y}_{k+j-1}^h + \mathbf{B}^h [\mathbf{y}^s + \boldsymbol{\mu}^y](\mathbf{u}_{k+j}) \quad (6.18b)$$

$$\mathbf{y}_k^h = \mathbf{y}_k^N \quad (6.18c)$$

$$\|\mathbf{u}_{k+j} - \mathbf{u}_k^{tg}\| \leq \Delta_k \quad (6.18d)$$

$$\mathbf{u}_{k+j} \in \mathbb{U} \quad (6.18e)$$

in which, N is the control horizon, $\mathbf{u}_k^{tg} := \mathbf{u}_k + \Delta \mathbf{u}_{k+1}^*$ are the input targets and $\mathbf{y}_k^{sp} := [\mathbf{y}^s + \boldsymbol{\mu}^y](\mathbf{u}_k^{tg})$ is the output setpoints calculated by the RTO, and \mathbf{y}_k^m is the vector of measured output variables; \mathbf{W}_y , \mathbf{W}_u and $\mathbf{W}_{\Delta u}$ are diagonal positive semidefinite weight matrices for the outputs, inputs and input variations, respectively; \mathbf{P} is the terminal cost calculated by a Lyapunov equation, see DELOU *et al.*

(2021c), which is possible since the nonlinear model is affine with the states and matrix A^h is forced to be stable at the identification stage.

6.4 Case Study: Williams-Otto Reactor

The Williams-Otto reactor problem is a commonly used system to assess MA techniques due to the presence of two models with structural differences, one to emulate the plant and the other to be used as the model. The plant reaction system comprises the following equations:



It consists of a continuous stirred-tank reactor (CSTR), in which reactants A and B are fed with mass flowrates F_A and F_B and reactor temperature T_R . The main goal is to produce components P and E with byproduct G . In the plant reaction system, there is still an intermediate component C .

To characterize the structural plant-model mismatch, the model reaction system does not include the intermediate component C nor its composition is measured. The two reactions of the model are:



The complete mass balance equations and parameter values for the plant and model are well detailed in ZHANG and FRASER FORBES (2000). The original optimization problem consists of maximizing a profit function with F_B and T_R as decision variables and constraints on the mass fractions of A and G . It can be mathematically expressed as:

$$\begin{aligned} \min_{F_A, T_R, \mathbf{x}} \quad & J := (1043.38w_P + 20.92w_E)(F_A + F_B) \\ & - 79.23F_A - 118.34F_B \end{aligned} \quad (6.21a)$$

$$\text{s.t. } \mathbf{0} = \mathcal{F}(\mathbf{x}, \mathbf{u}) \quad (6.21b)$$

$$g_1 := w_A - 0.12 \leq 0 \quad (6.21c)$$

$$g_2 := w_G - 0.08 \leq 0 \quad (6.21d)$$

$$F_B \in [4, 7], \quad (6.21e)$$

$$T_R \in [70, 100] \tag{6.21f}$$

in which, $\mathbf{x} := [w_A, w_B, w_E, w_P, w_G]^T$ is the state vector, function \mathcal{F} represents the model system of equations given by ZHANG and FRASER FORBES (2000) and $\mathbf{u} := [F_B, T_R]^T$.

In order to present results compatible with the work of DEL RIO CHANONA *et al.* (2021), the state vector was considered fully measured with an additive white noise with a standard deviation of 0.0005, and the RTO objective function was also measured with an additive white noise with a standard deviation of 0.5.

6.4.1 Comparison between MA-GP and MAy-GP

The trust-region algorithm is presented by DEL RIO CHANONA *et al.* (2021) and the parameters considered were: $\Delta_0 = 0.25$, $\Delta_{max} = 0.7$, $\eta_1 = 0.2$, $\eta_2 = 0.7$, $\gamma_{red} = 0.8$ and $\gamma_{inc} = 1.2$.

Figure 6.1 shows the comparison between the proposed MAy-GP, MA-GP, and no MA. Both MA-GP and MAy-GP present a similar behavior, preserving the ability to drive the plant to its true optimum. The result of MA-GP is compatible with DEL RIO CHANONA *et al.* (2021). The approach where no MA was used is only capable of driving the plant towards the model optimum, in this case study, the objective function value of the model optimum is quite near the plant optimum, but the difference is quite visible in Figure 6.1(d) analyzing the location of the inputs.

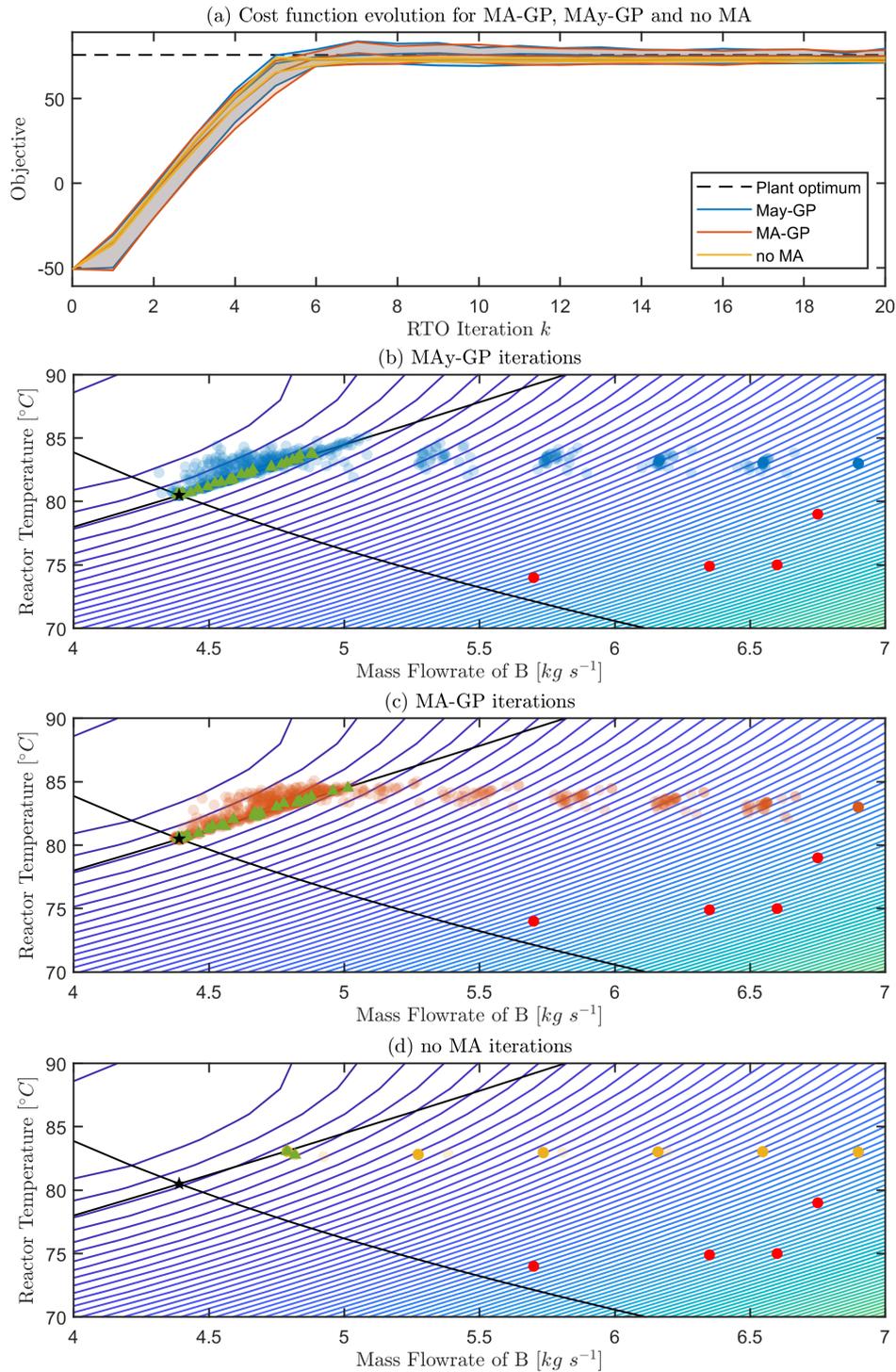


Figure 6.1: 20 RTO iterations for 30 noise realizations. The red dots are initial GP training points. Green triangles are the last iteration of each RTO run. Black lines are the constraint limits and the black star is the plant optimum.

All three approaches presented a fair variability. Variability in the case with no MA is explained only by measurement noise, that is the reason it is significantly inferior to the other two approaches. In MAy-GP and MA-GP, measurement noise is also responsible for a variation in the GP that is trained at each RTO iteration.

That is why its variability is significantly higher. This effect combined with the trust-region strategy are responsible for inserting a forced exploration effect on the GP training process, even though no extra exploration mechanism was added to the problem.

The last iterations of each RTO run in MAy-GP and MA-GP are spread between the model optimum and the plant optimum following the constraint border. To have a better visualization of these points, they were plotted in a histogram that can be visualized in Figure 6.2.

From the histogram chart, it is possible to see that, although some points are spread, most of them are located at or around the plant optimum in both MAy-GP and MA-GP. Therefore, MAy-GP presents similar potential as MA-GP in order to drive the plant towards its true optimal point.

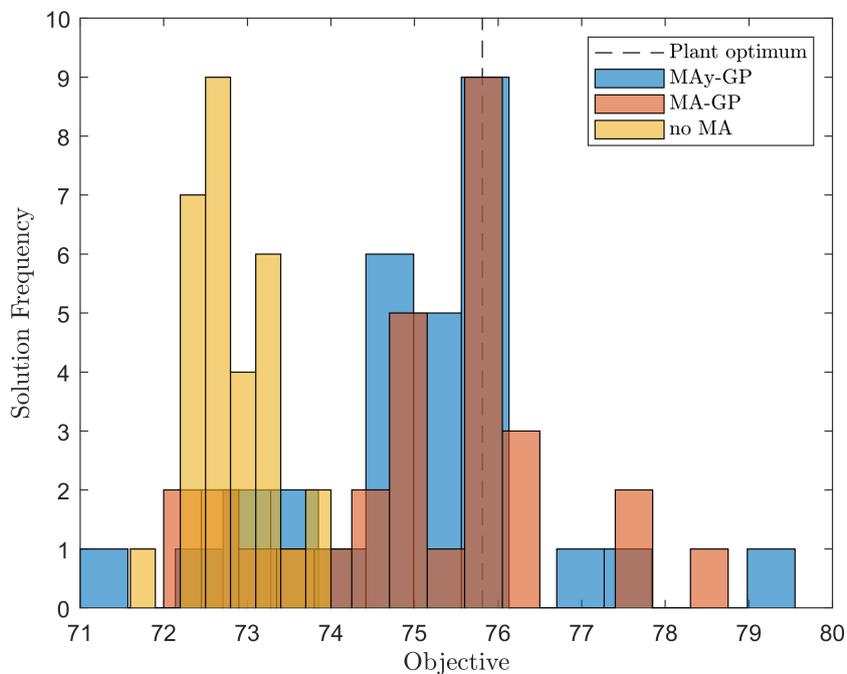


Figure 6.2: Histogram of the last iteration point of the 30 noise realization runs. The solution frequency axis shows how many solutions of the 30-noise realization runs stayed between each objective band.

6.4.2 NMPC with GP output modifiers

Figure 6.3 illustrates the dynamic evolution of the RTO objective value and the manipulated variables for two NMPC architectures, one considering GP output modifiers in the control and optimization layers, labeled as MAy-GP NMPC, and another considering only GP optimization modifiers, labeled MA-GP NMPC.

The NMPC tuning parameters considered were: $N = 3$, $\mathbf{W}_y = \mathbf{I}_{n_y}$, $\mathbf{W}_u = \mathbf{0}_{n_u}$

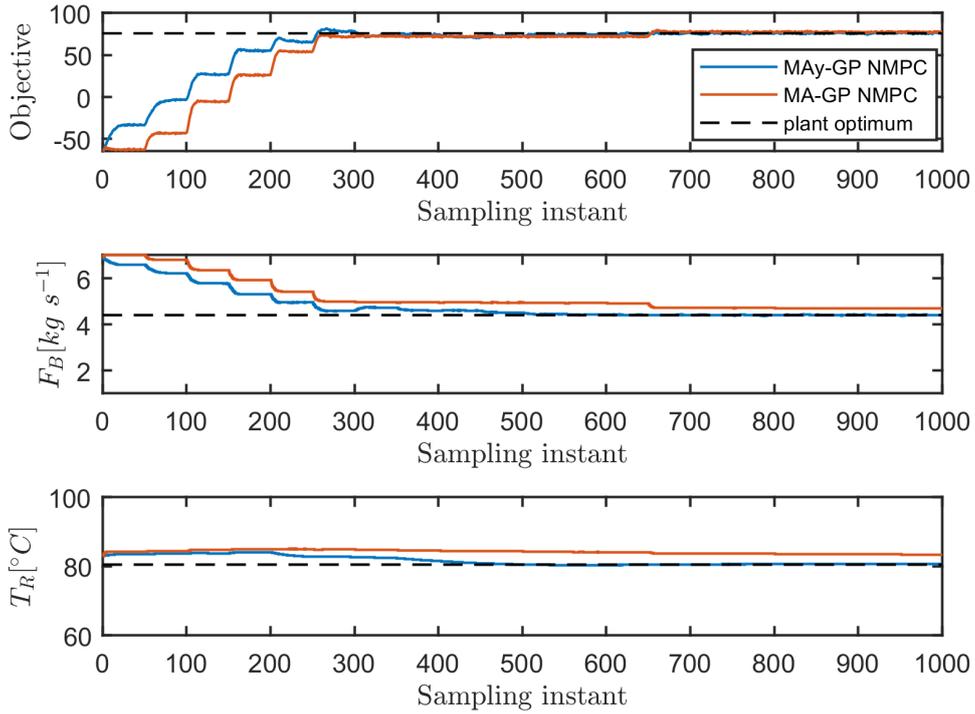


Figure 6.3: Dynamic evolution of the RTO objective function, and the manipulated variables, F_B and T_R .

and $\mathbf{W}_{\Delta u} = \text{diag}([10^{-1} \ 10^{-2}]^T)$, for both MAy-GP NMPC and MA-GP NMPC. The dynamic matrix \mathbf{A}^h was identified around a single operating point far from plant and model optima. The output setpoints and input targets were calculated by the RTO at 60 sampling instants period with a sampling time of 1 *min*.

The Hammerstein MAy-GP NMPC fairly drives the plant toward its optimal condition even though the predicted model used describes only a linear approximation of the dynamics with a structural mismatch in the static portion. This is made possible due to the GP correction terms added to the Hammerstein model output variables. This observation is supported by the fact that Hammerstein MA-GP NMPC is unable to drive the plant towards its true optimum due to the mismatch between the models of the control and optimization layers, showing that including GP correction terms only in the optimization problem is not enough to guarantee optimal operation. From another perspective, if MA-GP were chosen to be used in the optimization layer and MAy-GP in the control layer, the result would be practically the same as shown for MAy-GP NMPC, with only the additional effort of having to identify different modifiers for each layer.

Figure 6.4 shows the RTO objective function map over the manipulated variables for MAy-GP NMPC and MA-GP NMPC. It is possible to see the evolution of each NMPC iteration and the evolution of the trust regions. MAy-GP can fairly

seek the true plant optimum, keeping the system around it most of the time, which can be observed by the darker blue clouds, but MA-GP is unable to reach the true plant optimum, driving the system only towards a neighborhood of the model optimum.

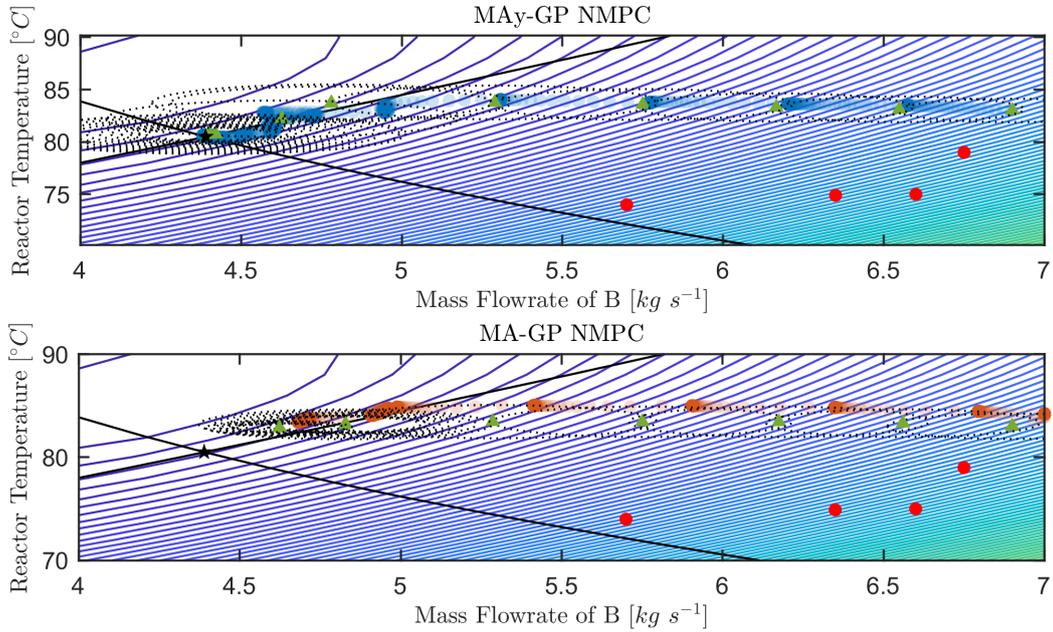


Figure 6.4: MAY-GP and Hammerstein NMPC iterations with trust regions evolution. The red dots are initial GP training points. Green triangles are the RTO solutions accepted by the trust-region algorithm and the black star is the plant optimum.

It is important to highlight that the RTO optimal input solution is not passed to the NMPC as targets, since the input weights are zero. This way, it is possible to assure that the MAY-GP NMPC tracks the optimal solution due to its great ability to predict the plant outputs due to the output GP corrections. Additionally, as the constraints are only applied in the steady-state optimization problem (Equation 6.21), in terms of system dynamics, it can evolve to unfeasible regions. The RTO solutions are within the plant feasible region, except for solution $[F_B, T_R] = [4.8, 83.7]$.

6.5 Conclusion

This chapter proposes an output Modifier adaptation based on Gaussian Process correction terms using the trust-region algorithm. The appealing aspect of using output MA instead of the classic MA is that the correction terms are interchangeable between several model-based techniques, such as controller and observers. It is shown that the performance of MA-GP and MAY-GP are similar, preserving the

ability to drive the plant towards its optimum even with a structural plant-model mismatch. We also propose a Hammerstein NMPC that can take advantage of the MAy-GP correction terms on its predictive model. It is shown that the NMPC using MAy-GP is able to drive the plant toward its optimal condition.

Chapter 7

A Comparison Between Process Control Strategies: Reinforcement Learning with RBFs and NMPC Coupled with EKF

A shorter version of this chapter was presented in the 23rd Brazilian Congress of Chemical Engineering (COBEQ), 2021, Gramado, Brazil, and published in the "Anais do 23° Congresso Brasileiro de Engenharia Química" ISSN: 2178-5600 (paper number 143150). Also, a version of this chapter was published in the Brazilian Journal of Chemical Engineering, June 2023 (DELOU *et al.*, 2023a).

7.1 Introduction

In the past few decades, the process industry has experienced a large increase in automation and data acquisition systems. Consequently, acquiring good quality and reliable data becomes more accessible, and data-based reinforcement learning (RL) methodologies become more susceptible to application. In an RL problem, an agent learns to realize a specific task by direct interaction with its environment with no supervision (SUTTON and BARTO, 2018b).

Another incentive for the application of RL methods arises from the drawbacks of classical model-based control. Most applications of model predictive control (MPC) methods in industry still rely on linear models (YU-GENG *et al.*, 2013), which fail to represent many chemical processes. The application of nonlinear MPC (NMPC) methodologies is progressing but still suffers from the following disadvantages: they require a nonlinear process model that may be too laborious to obtain, solving the NMPC optimization problem may be too computationally

demanding for online implementation, and the process degradation with time imposes that the model should be periodically updated to stay reliable, requiring the implementation of an adaptive mechanism.

For the implementation of RL techniques in a real process, two approaches are proposed in SHIN *et al.* (2019a):

- The integration of RL and MPC in a unique framework to combine the benefits of both techniques.
- The direct substitution of current process control technologies with RL.

The latter is considered in this chapter. In that case, the training of the agent through interaction with the real system is not acceptable as it would lead to unproductive and unsafe operation for a great amount of time. That is the reason why a two-step approach needs to be considered for the implementation of RL techniques in process control: In the first step, a simulated environment together with real process data, if available, are used for training to obtain a good enough controller for online implementation. In the second step, the controller obtained in the first step is calibrated in the real process.

An RL actor-critic-based controller using linear combinations of RBFs for the value function and policy approximations is then proposed for the control and optimization of the Van de Vusse (VdV) reactor, a challenging benchmark well-known for its nonlinear dynamics and change of the sign of the process gain. A sensitivity analysis on some of the training parameters is done to guide the training process, and the performance of the obtained RL controller is compared with an NMPC coupled with an extended Kalman filter (EKF). This latter controller is used as a reference given its good performance in mixing optimization and reference tracking, and in handling unmeasured disturbances. Note that in CASSOL *et al.* (2018b), the authors also compared the performance of an RL-based controller with an NMPC controller for the control and optimization of the VdV reactor. However, a value-based algorithm was used for training, a neural network was used to approximate the value function, and its performance was compared to an NMPC controller for which no state estimation mechanism was implemented to deal with unmeasured disturbances.

The remainder of this chapter is organized as follows: In Section 7.2, the RL-based controller using linear combinations of RBFs for the policy and value function approximations is described. In Section 7.3, the NMPC controller coupled with EKF used as a reference is presented. In Section 7.4, the CSTR with the Van de Vusse reaction used as a benchmark and the corresponding control problem are introduced. In Section 7.5, a sensitivity analysis on some of the parameters used for the RL controller training is done, and the performances of the RL-based

and NMPC controllers are compared. In Section 7.6, the chapter is concluded. Finally, an in-depth description of the RL problem is presented in Appendix D.

7.2 The RL-RBF controller

In the application of RL techniques to process control, the agent is the controller, and the process the environment it interacts with. This section presents the procedure used for the training of the RL-RBF controller proposed in this work, where the episodic actor-critic algorithm with eligibility traces described in SUTTON and BARTO (2018b) is implemented. Linear combinations of Gaussian RBFs are used to approximate both value function and policy. The value function is updated online with the use of the backward view $TD(\lambda)$ method presented in section D adapted to continuous MDPs, and the policy is updated using the gradient ascent approach towards the value function maximization.

Consider a process represented by a state vector \mathbf{s} of n_s variables and an action vector \mathbf{a} of n_a variables manipulated by the RL-RBF controller. The value function and policy associated with a state $\mathbf{s} \in \mathbb{R}^{n_s}$ are given by Equation 7.1 and 7.2, respectively.

$$V(\mathbf{s}) = \boldsymbol{\theta}_V^T \boldsymbol{\Phi}(\mathbf{s}) \quad (7.1)$$

$$\boldsymbol{\pi}(\mathbf{s}) = \boldsymbol{\Theta}_\pi^T \boldsymbol{\Phi}(\mathbf{s}) \quad (7.2)$$

in which $\boldsymbol{\Phi}(\mathbf{s})$ represents a vector of n_c state-dependent Gaussian RBFs defined by Equation 7.3, $\boldsymbol{\theta}_V \in \mathbb{R}^{n_c}$ and $\boldsymbol{\Theta}_\pi \in \mathbb{R}^{n_c \times n_a}$ are the learned parameters for the value function and the policy, respectively.

$$\boldsymbol{\Phi}(\mathbf{s}) = e^{-\|\mathbf{s}-\mathbf{c}\|_{\mathbf{W}}^2} \quad (7.3)$$

in which $\mathbf{W} \in \mathbb{R}^{n_s \times n_s}$ is used to adjust the width of the Gaussian bell-shaped functions, and \mathbf{c} is the vector of n_c Gaussian RBF centers considered in the state space.

With a continuous state space, an eligibility trace cannot be assigned to each state. Instead, a short-term memory vector $\mathbf{z} \in \mathbb{R}^{n_c}$ initialized to $\mathbf{0}$ at the beginning of each learning episode parallels the weight vector $\boldsymbol{\theta}_V \in \mathbb{R}^{n_c}$. The idea behind this mechanism is that when a component of $\boldsymbol{\theta}_V$ participates in producing a new estimate of the value function, the corresponding component of \mathbf{z} is increased by the corresponding element of $\boldsymbol{\Phi}(\mathbf{s})$ before it begins to fade away. Learning will then occur in that component of $\boldsymbol{\theta}_V$ if a nonzero TD error occurs before the trace

falls back to zero (SUTTON and BARTO, 2018b).

Before training, the policy and value function parameters, respectively Θ_π and θ_V , are initialized. At the beginning of each training episode, the eligibility trace $\mathbf{z} \in \mathbb{R}^{n_c}$ is initialized to $\mathbf{0}$, and the state is initialized. At each episode step, the actual state \mathbf{s} is communicated to the actor that computes the action \mathbf{s} to be implemented in the environment according to Equation 7.4.

$$\mathbf{a} = \Theta_\pi^T \Phi(\mathbf{s}) + \boldsymbol{\nu} \quad (7.4)$$

in which $\boldsymbol{\nu} \in \mathbb{R}^{n_a}$ is an exploration noise sampled from a zero mean and variance σ^2 normal probability distribution. The variance σ^2 decreases as the agent is being trained, as shown in Equation 7.5.

$$\sigma^2 = \frac{N_0}{N_0 + N(\mathbf{s})} \quad (7.5)$$

in which $N(\mathbf{s})$ is the number of times the closest center to \mathbf{s} has been visited. Upon action \mathbf{a} the environment evolves to the next state $\mathbf{s}' \in \mathbb{R}^{n_s}$. From the immediate reward $r \in \mathbb{R}$ obtained from this transition and Equation 7.1, the TD error defined in Equation D.12 is computed using Equation 7.6.

$$\delta = r + \gamma \theta_V^T \Phi(\mathbf{s}') - \theta_V^T \Phi(\mathbf{s}) \quad (7.6)$$

And the eligibility trace is updated through Equation 7.7.

$$\mathbf{z} \leftarrow \gamma \lambda \mathbf{z} + \Phi(\mathbf{s}) \quad (7.7)$$

Then, the critic and the actor update θ_V and Θ_π , respectively, using Equation 7.8.

$$\begin{aligned} \theta_V &\leftarrow \theta_V + \alpha_V \mathbf{z} \delta \\ \Theta_\pi &\leftarrow \Theta_\pi + \alpha_\pi \Phi(\mathbf{s}) \delta (\mathbf{a} - \Theta_\pi^T \Phi(\mathbf{s}))^T \end{aligned} \quad (7.8)$$

in which $\alpha_V \in \mathbb{R}$ and $\alpha_\pi \in \mathbb{R}$ are the learning rates for θ_V and Θ_π , respectively. Finally, \mathbf{s}' becomes the new actual state \mathbf{s} , and the above procedure can be implemented again. The RL-RBF controller training algorithm was developed in MATLAB R20215a.

7.3 The NMPC-EKF controller

An NMPC controller coupled with EKF for state estimation is considered a reference given its good performance in mixing optimization and reference tracking, and in handling unmeasured disturbances. Consider a process with nx states, ny controlled outputs, and nu manipulated inputs. A model for this process is given in Equation 7.9.

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \end{aligned} \quad (7.9)$$

in which $\mathbf{x}_k \in \mathbb{R}^{nx}$, $\mathbf{u}_k \in \mathbb{R}^{nu}$, and $\mathbf{y}_k \in \mathbb{R}^{ny}$ represent the process state, manipulated, and controlled variables at time instant k , respectively.

In the NMPC framework, from the current state estimation $\hat{\mathbf{x}}_k \in \mathbb{R}^{nx}$, the sequence of N control moves represented by the vector $\Delta \mathbf{u} \in \mathbb{R}^{N \times nu}$ that optimizes the open-loop prediction of the L future states obtained through the integration of the process model defined in Equation 7.9 is computed, where N and L are the control and prediction horizons, respectively. According to the receding horizon control paradigm, only the first input is implemented in the process until the next sampling time is reached. Then, the new state is estimated, and a new control sequence is calculated.

The optimization problem solved at each sampling time k is defined in Equation 7.10.

$$\begin{aligned} \min_{\Delta \mathbf{u}} & \sum_{i=1}^L \|\mathbf{y}_{k+i} + \boldsymbol{\epsilon}_k - \mathbf{y}_{k+i}^{SP}\|_{\mathbf{W}_y}^2 + \sum_{i=0}^{N-1} \|\Delta \mathbf{u}_{k+i}\|_{\mathbf{W}_{\Delta u}}^2 \\ \text{s.t.} \quad & \mathbf{x}_{k+j} = \mathbf{f}(\hat{\mathbf{x}}_{k+j-1}, \mathbf{u}_{k+j-1}) \\ & \mathbf{y}_{k+j} = \mathbf{h}(\hat{\mathbf{x}}_{k+j-1}, \mathbf{u}_{k+j-1}) \\ & \mathbf{x}_k = \hat{\mathbf{x}}_k \\ & -\Delta \mathbf{u}_{max} < \Delta \mathbf{u}_{k+j} < \Delta \mathbf{u}_{max} \\ & j = 1, 2, \dots, L \end{aligned} \quad (7.10)$$

where $\boldsymbol{\epsilon}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k$ represents the mismatch between the measurement and estimate of the plant output, $\mathbf{W}_y \in \mathbb{R}^{ny \times ny}$ and $\mathbf{W}_{\Delta u} \in \mathbb{R}^{nu \times nu}$ are weighing matrices, and $\Delta \mathbf{u}_{max} \in \mathbb{R}^{nu}$ is the control move upper limit.

In this work, the states are estimated through the use of an Extended Kalman Filter, adapted to systems with nonlinear dynamics. At time instant $k + 1$ the state

is estimated by following the steps shown in Equation 7.11 (SIMON, 2006a):

$$\begin{aligned}
\mathbf{A}_k &= \frac{d\mathbf{f}}{d\mathbf{x}_k}(\hat{\mathbf{x}}_k, \mathbf{u}_k) \\
\mathbf{C}_k &= \frac{d\mathbf{h}}{d\mathbf{x}_k}(\hat{\mathbf{x}}_k) \\
\mathbf{K}_k &= \mathbf{P}_k \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^T + \mathbf{R}_k)^{-1} \\
\hat{\mathbf{x}}_{k+1} &= \mathbf{f}(\hat{\mathbf{x}}_k, \mathbf{u}_k) + \mathbf{K}_k [\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k)] \\
\mathbf{P}_{k+1} &= \mathbf{A}_k (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k
\end{aligned} \tag{7.11}$$

where $\mathbf{K}_k \in \mathbb{R}^{n_x \times n_y}$ is the Kalman Filter gain, $\mathbf{P}_k \in \mathbb{R}^{n_x \times n_x}$ is the estimation-error covariance, $\mathbf{Q}_k \in \mathbb{R}^{n_x \times n_x}$ is the covariance of the process noise, and $\mathbf{R}_k \in \mathbb{R}^{n_y \times n_y}$ is the covariance of the measurement noise.

The NMPC algorithm was developed in CasADi (ANDERSSON *et al.*, 2019) with the method of polynomial approximation in finite elements (BIEGLER, 1984b) for the profiles of states and manipulated variables in time.

7.4 The Van de Vusse reactor Benchmark

7.4.1 The process

The performance of the proposed controller is tested through the simulation of the control and optimization of a non-isothermal CSTR, with the Van de Vusse reaction, as described in KLATT and ENGELL (1998). The reaction consists of the synthesis of cyclopentenol (B) from cyclopentadiene (A) by acid-catalyzed electrophilic addition of water in a dilute solution. Due to the strong reactivity of both A and B, dicyclopentadiene (D) is produced by the Diels-Alder reaction as a side product and cyclopentanediol (C) as a consecutive product by the addition of another water molecule. The corresponding reactions are represented in Equation 7.12.



It is usually desirable to maximize the production of component B while minimizing the production of C and D. The system dynamics is modeled by the mass balances of A and B, and the energy balance represented by Equations 7.13, 7.14, and 7.15, respectively.

$$\frac{dC_A}{dt} = \frac{F}{V} (C_{A,in} - C_A) - k_1(T)C_A - k_3(T)C_A^2 \tag{7.13}$$

$$\frac{dC_B}{dt} = \frac{F}{V}(C_{B,in} - C_B) + k_1(T)C_A - k_2(T)C_B \quad (7.14)$$

$$\begin{aligned} \frac{dT}{dt} = \frac{1}{\rho c_p} [k_1(T)C_A(-\Delta H_1) + k_2(T)C_B(-\Delta H_2) + k_3(T)C_A^2(-\Delta H_3)] \\ + \frac{F}{V}(T_{in} - T) + \frac{k_w A_R}{\rho c_p V}(T_k - T) \end{aligned} \quad (7.15)$$

where the rate constants $k_i(T)$ are given by the Arrhenius law in Equation 7.16.

$$k_i(T) = k_{i,0} e^{(E_i/T)} \quad (7.16)$$

System parameters can be found in TRIERWEILER (1997), and are reproduced in Table 7.1.

Table 7.1: Van de Vusse reactor parameters

Parameter	Value	Unit
$k_{1,0}$	1.287×10^{12}	h^{-1}
$k_{2,0}$	1.287×10^{12}	h^{-1}
$k_{3,0}$	9.043×10^9	$L/mol/h$
E_1	-9758.3	K
E_2	-9758.3	K
E_3	-8560	K
ΔH_1	4.2	kJ/mol
ΔH_2	-11	kJ/mol
ΔH_3	-41.85	kJ/mol
V	10	L
C_p	3.01	$kJ/kg/K$
ρ	0.932	kg/L
A_R	0.215	m^2
k_W	4032	$kJm^{-2}/h/K$

The Van de Vusse CSTR system is a very well-known benchmark for process system engineering case studies. This process was chosen due to its nonlinear characteristics, and in particular the change of the sign of the gain of the product concentration C_B with respect to inlet flow F as illustrated in Figure 7.1. The location of the inversion depends on the reactor temperature T , which in turn is controlled by manipulating the cooling jacket temperature T_k .

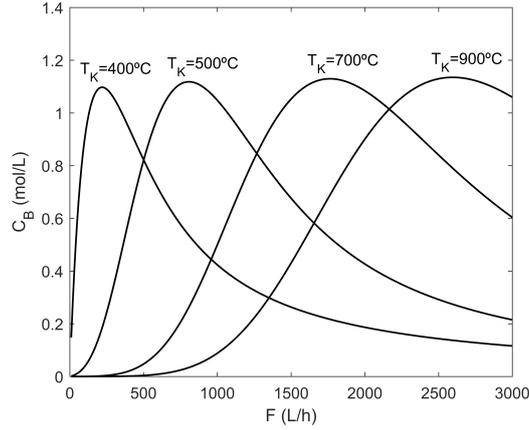


Figure 7.1: Open-loop non-minimal phase response of the Van de Vusse CSTR.

7.4.2 Definition of the control problem

The control objective is to track a setpoint for F_B at $F_B^{SP} = 2000 \text{ mol/h}$ while maximizing C_B by manipulating the inlet flow F , and the cooling jacket temperature T_k . The performance of the controllers of this work are compared using a 120-hour simulation with a sampling time set to $\Delta t = 0.01 \text{ h}$. At the beginning of the simulation, the process is at the steady state defined by the conditions shown in Table 7.2.

Table 7.2: Initial conditions

Variable	Value	Unit
$C_{A,in}$	5.1	mol/L
$C_{B,in}$	0	mol/L
T_{in}	100	$^{\circ}\text{C}$
F	1000	L/h
T_K	100	$^{\circ}\text{C}$
C_A	3.856	mol/L
C_B	0.6561	mol/L
T	119.488	$^{\circ}\text{C}$

The inlet reagent concentrations $C_{A,in}$ and $C_{B,in}$ are fixed during the simulation while T_{in} is changed to 90°C at $t = 60 \text{ h}$, representing an unmeasured disturbance. Finally, the manipulated variable increments ΔF , and ΔT_k at each sampling time are limited to $\pm 1.2 \text{ L/h}$, and $\pm 0.48 \text{ K}$, respectively.

7.5 Results

7.5.1 Training of the RL-RBF Controller

In this work, $\mathbf{s} = [C_B, C_A, T, F, T_K]^T$ is the vector of states representing the process, and $\mathbf{a} = [\Delta F, \Delta T_K]$ is the vector of action variables defined as increments of the process manipulated variables. In any RL problem, the choice of the reward function is very important as it is the only element through which the user can communicate to the agent what it has to accomplish. In this work, the reward function is defined in Equation 7.17.

$$r_k = |F_B(k-1) - F_B^{SP}| - |F_B(k) - F_B^{SP}| + w(C_B(k) - C_B(k-1))[1 - \tanh(|F_B(k) - F_B^{SP}|/\epsilon)] \quad (7.17)$$

The first two terms of the equation reward the decrease of the distance between F_B and its setpoint within the sampling time, and the last term rewards the increase on C_B , aiming for its maximization. The weight w is a scaling parameter, and the hyperbolic function works as a gradual activator of the optimization term when F_B reaches its setpoint, with the parameter ϵ controlling the gradient of this function. The main difference between this approach to the one used by CASSOL *et al.* (2018b) is that the authors proposed a logistic activator function which is very soft around its maximum, while the hyperbolic function is stiffer.

For the value function and policy approximations, the positions of the Gaussian RBF center for each state variable is presented in Table 7.3. The matrix \mathbf{W} that

Table 7.3: Chosen centers for each state variable.

State Variable	Centers	Unit
C_B	$[0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1 \ 1.2]^T$	mol/L
C_A	$[0 \ 1.7 \ 3.4 \ 5.1]^T$	mol/L
T	$[270 \ 350 \ 430 \ 510]^T$	K
F	$[500 \ 1000 \ 1500 \ 2000 \ 2500]^T$	L/h
T_k	$[270 \ 785 \ 1300]^T$	K

defines the width of the RBFs was set to $\mathbf{W} = \text{diag}([100, 1.3841, 6.25 \times 10^{-4}, 4 \times 10^{-6}, 1.5082 \times 10^{-5}])$.

The tuning parameter values associated with the actor-critic algorithm presented in D are presented in Table 7.4.

Table 7.4: RL tuning parameters values.

γ	λ	α_V	$\alpha_{\pi,1}$	$\alpha_{\pi,2}$
0.9	0.5	$1 \cdot 10^{-1}$	$5 \cdot 10^{-2}$	$5 \cdot 10^{-2}$

In the training stage, episodes of 6000 sampling times are considered. The inlet properties are fixed at $C_{A,in} = 5.1$ mol/L, $C_{B,in} = 0$ mol/L, and $T_{in} = 100^\circ\text{C}$. At each episode, the initial values for F and T_k are sampled from a uniform distribution over $[500, 2500]$ L/h and $[23, 723]$ $^\circ\text{C}$ and are used to compute the episode's initial steady state.

In the following sections, a sensitivity analysis on the number of training episodes N_{ep} (section 7.5.1.1), the exploration through parameter N_0 (section 7.5.1.2), the activation function width used in the reward through parameter ϵ (section 7.5.1.3), and the weight on C_B maximization w also used in the reward function (section 7.5.1.4) is done to guide the RL-RBF controller training. The default values for these parameters are shown in Table 7.5

Table 7.5: Default training parameters for the sensitivity analysis

N_{ep}	N_0	ϵ	w
50	100000	150	2500

7.5.1.1 Effect of the number of training episodes

The performances of RL-RBF controllers trained with 10, 30, 50, and 70 episodes are shown in Figure 7.2. It can be observed that the performance of the controller improves with the number of episodes used for training. The more the controller interacts with the simulated process the more data, and consequently, the more information is available for training. With ten episodes, the setpoint for F_B is not even reached within the simulation time. From 30 episodes on, the performance becomes satisfactory as F_B reaches its setpoint, and concentration C_B is maximized. The best performances are obtained with 50 and 70 episodes with faster responses than with 30 episodes. The small performance difference observed between these two simulations indicates that there is several episodes from which the controller stops learning.

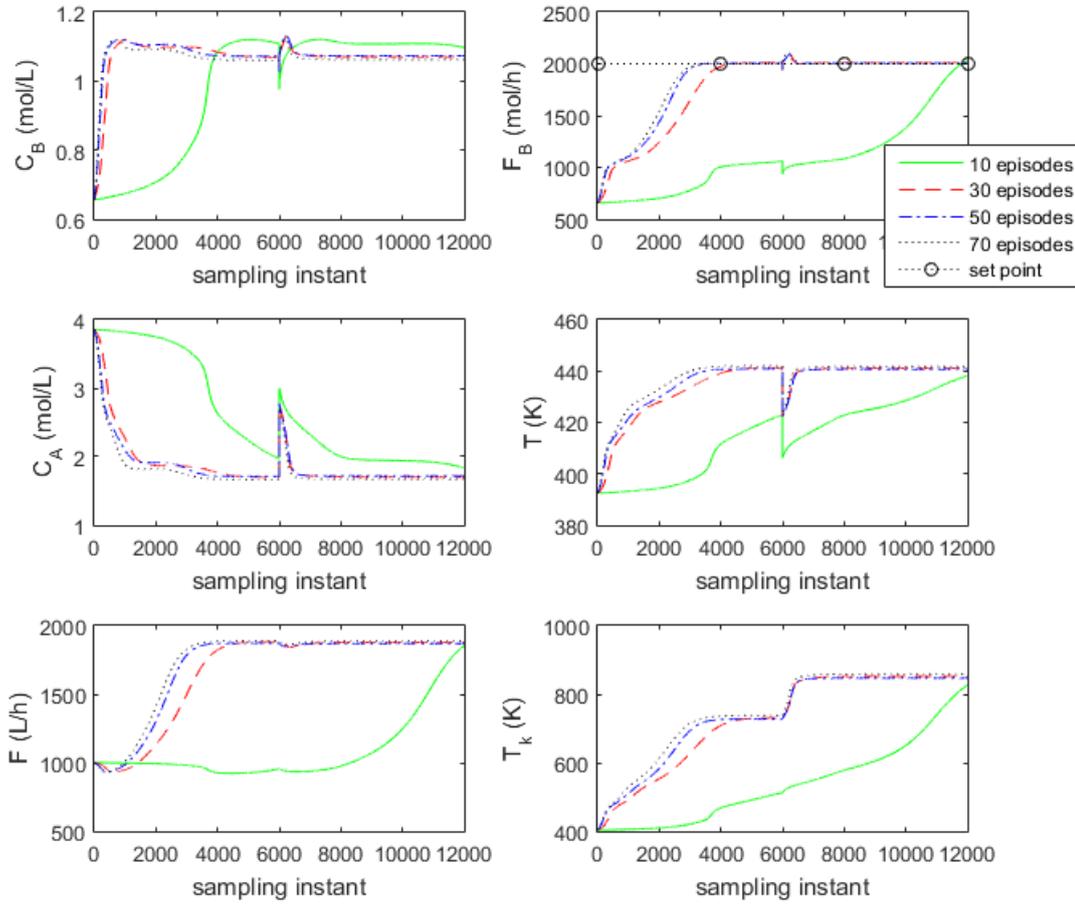


Figure 7.2: Influence of the number of episodes on training.

7.5.1.2 Effect of the training exploration

The influence of exploration on the performance of the RL-RBF controller can be observed in Figure 7.3, where the constant N_0 used in Equation 7.5 to define the decrease rate of the exploration noise variance σ^2 is increased from 10^3 to 10^9 . The higher this value, the slower σ^2 decreases, and the more the state space is explored by the controller. It can be observed that the higher the exploration, the better the performance. When exploration is the lowest ($N_0 = 10^3$), the controller does not manage to bring F_B to its setpoint within the simulation time, clearly indicating that more exploration is needed. From $N_0 = 10^5$ on, satisfactory responses are observed. The best performance is obtained for the intermediate value of $N_0 = 10^6$. With $N_0 = 10^9$, exploration has not been decreased enough at the end of the training to allow the convergence to the best policy, as commented in Appendix D. To obtain at least as good results as with $N_0 = 10^6$, the number of training episodes would have to be increased to guarantee that exploration has sufficiently

been reduced at the end of the training.

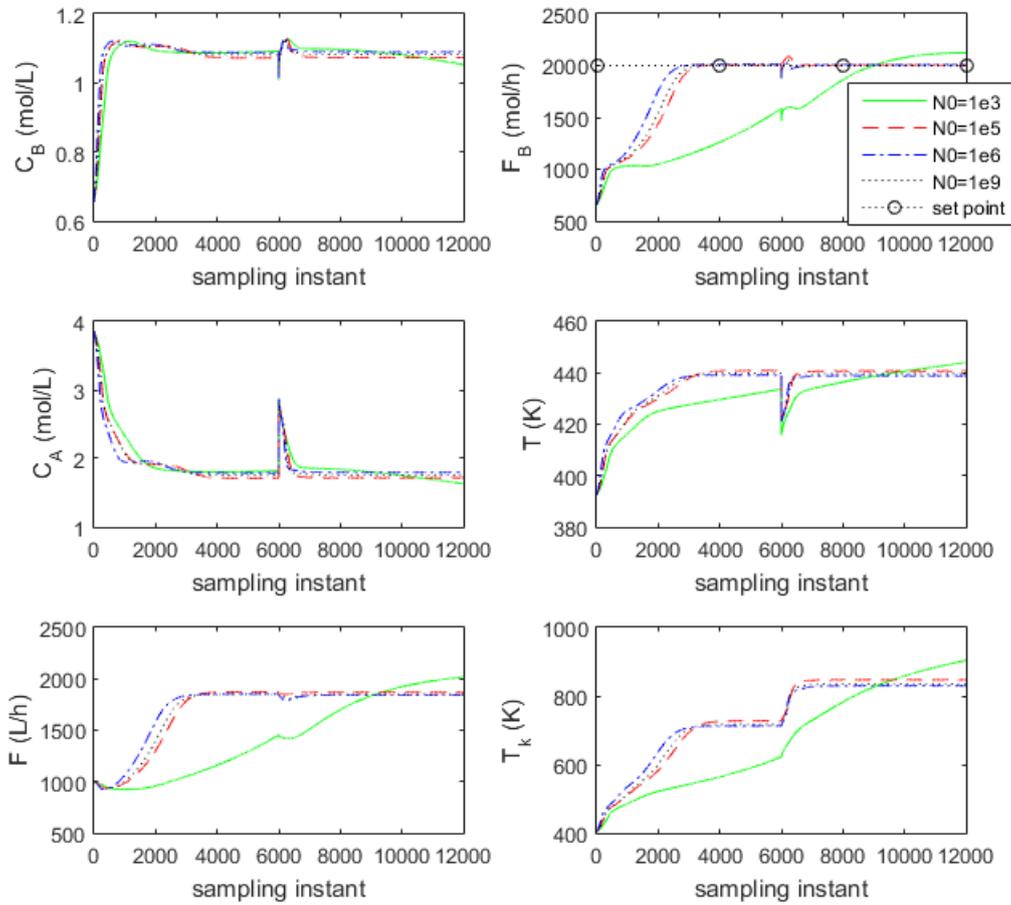


Figure 7.3: Influence of exploration on training.

7.5.1.3 Effect of the concentration maximization activation function width used in the reward function

The influence of the activation function width ϵ used in the reward defined in Equation 7.17 on the performance of the RL-RBF controller can be observed in Figure 7.4. It can be observed that the performance of the RL-RBF controller improves as ϵ increases. The results are very similar for $\epsilon = 150$ and $\epsilon = 300$, indicating that the performance stops getting better from a certain value of ϵ . Also, the very poor performance obtained with $\epsilon = 5$ shows that an excessively stiff activation function makes the maximization of C_B too difficult with this number of episodes.

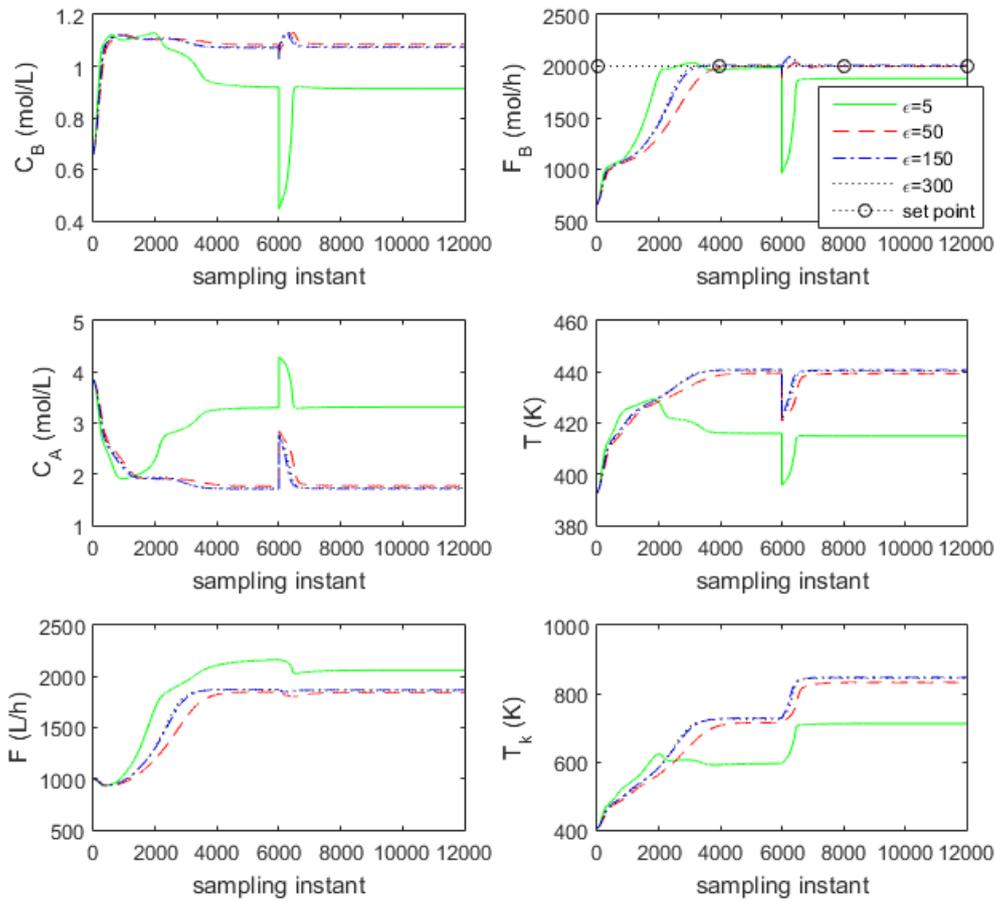


Figure 7.4: Influence of the concentration maximization activation function width used in the reward function.

7.5.1.4 Effect of the weight on concentration maximization in the reward function

The influence of the weight on C_B maximization w in the reward function defined in Equation 7.17 on the performance of the RL-RBF controller can be observed in Figure 7.5. When it is too low, as for the case where $w = 0.01$, the maximization of B is clearly neglected, and C_B is not maximized. On the contrary, when it is too high, as for the case where $w = 10000$, too much weight is put on the maximization, and an offset is obtained for F_B . An optimum then exists between these two values and the best performance was obtained for $w = 2500$.

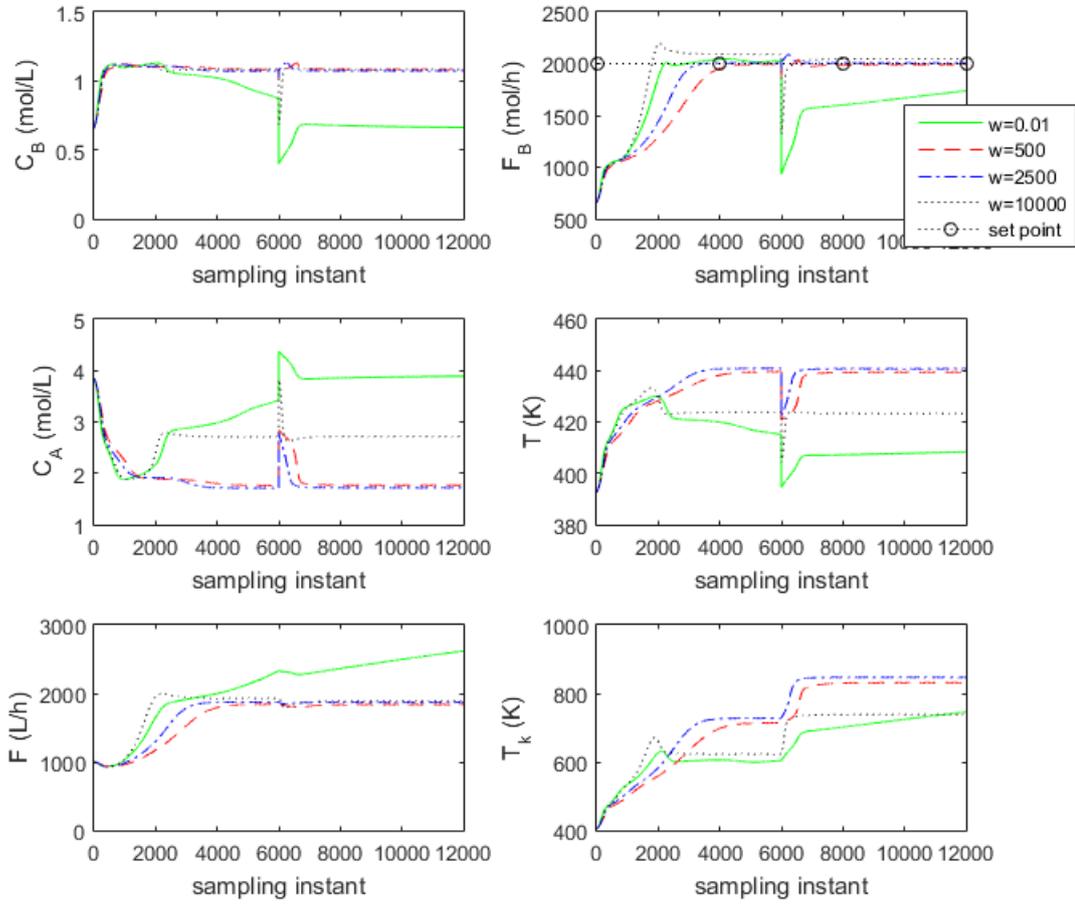


Figure 7.5: Effect of the weight on concentration maximization in the reward function.

7.5.2 NMPC-EKF controller settings

In the control optimization problem defined in Equation 7.10, the vector of states variables is $\mathbf{x} = [C_A, C_B, T]^T$, the vector of controlled variables is $\mathbf{y} = [C_B, F_B]^T$, and the vector of manipulated variables is $\mathbf{u} = [F, T_k]^T$. The setpoint for \mathbf{y} is set to $\mathbf{y}^{SP} = [2000, 1, 5]^T$ where 1, 5 is an unreachable upper value for C_B meant to allow its optimization. The weighing matrices were fixed at $\mathbf{W}_y = \text{diag}([1, 10^{-5}])$ and $\mathbf{W}_{\Delta u} = \text{diag}([5 \times 10^{-9}, 10^{-5}])$. The prediction horizon is set at $L = 12$, and the control horizon at $N = 3$. Finally, the upper bound on the control move effort is set at $\Delta \mathbf{u}_{max} = [1.2, 0, 48]^T$.

For the state estimation using EKF, the covariance of the process noise \mathbf{Q} and the covariance of the measurement noise \mathbf{R} were set to $\text{diag}([10^{-4}, 10^{-4}, 10^{-4}])$ and $\text{diag}([10, 10, 4000])$, respectively. The estimation-error covariance \mathbf{P}_k is initialized at $\text{diag}([1000, 1000, 1000])$.

7.5.3 Performance comparison between the NMPC-EKF and the RL-RBF controllers

In this section, the performances of the NMPC-EKF controller defined in Section 7.3 and the RL-RBF controller presented in Section 7.2 are compared. The training of the RL-RBF controller was guided by the sensitivity analysis done through Sections 7.5.1.1, 7.5.1.2, 7.5.1.3, and 7.5.1.4. The corresponding training parameters that were chosen for the RL-RBF controller are gathered in Table 7.6. The results

Table 7.6: Training parameters for the RL-RBF controller

N_{ep}	N_0	ϵ	w
500	10^8	150	2500

of the comparison are presented in Figure 7.6.

It can be observed that both controllers successfully manage to bring F_B to its setpoint, and C_B to its maximum. The performances are also similar. While, in the first part of the simulation, the NMPC-EKF controller presents a slightly faster response, the RL-RBF controller is better at compensating for the disturbance on the inlet temperature introduced at sampling time 6000. This result shows that for the control and optimization of the VdV reactor, the use of linear combinations of Radial Basis Functions to represent the value function, and the policy in an RL algorithm is a good alternative to neural networks with the advantage of guaranteeing the convergence of the algorithm.

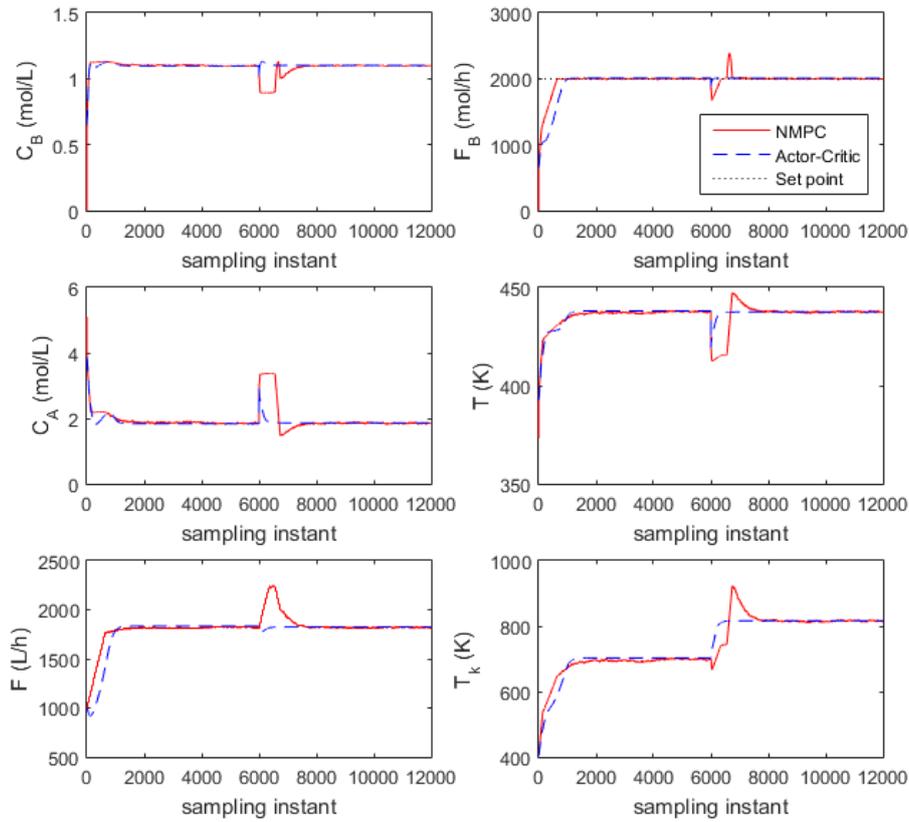


Figure 7.6: Performance comparison between RL with RBF approximation and NMPC.

7.6 Conclusion

In this work, the use of linear combinations of RBFs for policy and value function approximations in an RL actor-critic-based controller was assessed. A sensitivity analysis of some of the training parameters was done to guide the controller training process. It was shown that: exploration can always be increased as long as the number of episodes is increased accordingly; the concentration maximization activation function cannot be excessively stiff; and there is an optimal intermediate value for the weight on the concentration maximization in the reward function. The proposed RL-RBF controller was compared to an NMPC coupled with EKF for the control and optimization of the VdV reactor. It was shown that both controllers successfully managed to attain the control objectives under unmeasured disturbances with minor differences in performance. This study showed that the use of linear combinations of RBFs for policy and value function approximations is a successful alternative to neural networks, with the advantage of guaranteeing the convergence of the RL algorithm.

Chapter 8

Final Remarks

The present thesis explores the application of the HRTO methodology in the absence of a rigorous dynamic process model. An adequate dynamic approximate model that benefits from the available static model was proposed in order to enable the use of HRTO with low effort in current RTO methodologies.

Initially, in Chapter 3, several HRTO architectures were proposed using the dynamic approximate model. It was shown the proposed methodology preserved the observability characteristics of the static model and resulted in an adequate dynamic model to be used in an RTO framework. The proposed architectures explored a fixed dynamic model and two adaptive schemes, one with simultaneous estimation of the process parameter and disturbances together with the parameters of the dynamic model in the EKF formulation, and the other that separates the estimation of the process parameter and disturbances in the EKF formulation and the parameters of the dynamic model in a RELS approach. However, the most adequate approach was the one using a fixed model, since the proposed adaptation strategies presented no mechanism to prevent the arising of unstable and undesirable oscillatory modes.

In Chapter 4, a new and complete HRTO and control framework was proposed. The strategy relied on the previously proposed dynamic approximation model used in the EKF formulation and a new Self-Optimizing Infinite Horizon MPC formulation. The following characteristics of the proposed framework are highlighted:

- Use of HRTO in the absence of a reliable first-principles dynamic model;
- Use of stabilizing constraint in the identification stage to ensure the stability of the Hammerstein model structure;
- Inclusion of the economic objectives into the controller layer through the self-optimizing variables, which allows a more reliable and robust control layer;

- New proposition for naturally handling both types of active set changes: when a new constraint becomes active and when an active constraint becomes inactive;
- No wait time between optimization runs guarantees that the set of active constraints is always known;
- There is full compatibility between the models used in the observer, controller, and optimizer.

The results presented in the Williams-Otto reactor in the presence of parametric uncertainty showed that the proposed framework outperformed the framework that uses SSD in a classic RTO formulation. However, both frameworks presented the same ability to drive the plant to its optimum at a steady state. Moreover, the proposed framework presented improved economics, especially in a scenario of change in the active constraints.

In Chapter 5, we implemented two simplified versions of the previously proposed Hammerstein HRTO/ROPA to a lab-scale experimental rig that emulates a subsea oil well network. The first version considers the originally proposed Hammerstein model, where a prior identification step is performed based on plant data. And, the second version explores the assumption that the static model can be used directly in the EKF algorithm when the plant dynamics are sufficiently fast. In this case, the static model is used as an instantaneous dynamic model. Some general guidelines for the Hammerstein dynamic matrix practical identification are provided. The performance of both HROPAs was compared to a previous implementation of ROPA in the same experimental rig under the same disturbance scenario. It was shown that both HROPAs presented similar operational profiles to the classic ROPA, preserving the ability to observe the uncertain parameters, but with an expected slightly lower economic performance. The lower economic performance is justified not only by the use of dynamic approximations but also by the lack of EKF fine-tuning and changes to the experimental conditions. Therefore, it is fair to say that HROPA can produce a similar performance to ROPA with a considerably lower effort in the dynamic model design stage, establishing the basis for future applications of HROPA in large-scale systems

In Chapter 6, the assumption of an available adequate static model was dropped, and structural plant-model mismatch is explored. The previously proposed MA-GP was extended by proposing an output MA-GP (MAy-GP) methodology using trust-region concepts. Both of the techniques, MA-GP and MAy-GP, present the ability to drive the plant towards its true optimal point even in the presence of structural uncertainty, but MAy-GP presents the benefit of having

interchangeable correction terms to other model-based techniques, such as controllers and observer, while the MA-GP correction terms are designed only for a single optimization problem. Therefore, MAy-GP is a more flexible approach due to the fact that it corrects the model outputs instead of correcting specific objective function and constraints. Moreover, we propose a new NMPC formulation based on the approximate Hammerstein dynamic model taking advantage of the MAy-GP correction terms on the static part of its predictive model. It was shown that the framework considering the MAy-GP correction terms in both controllers and optimization layers was able to drive the plant towards its true optimal, but the one considering MA-GP in the optimization layer and no correction in the control layer was not capable. The proposed framework presented a way to deal with the static model when there is a considerable plant-model mismatch. The Hammerstein framework is used in the control layer to enable full compatibility between optimization and control. However, the framework did not include a way to estimate the MAy-GP correction terms dynamically, therefore SSD is still required to update the correction terms. A unified HRT0 with MAy-GP correction terms is a natural future work since no parametric adaptation was considered simultaneously to the modifier adaptation approach.

Finally, Chapter 7 explored the incorporation of the economic goals directly in a feedback-optimizing control strategy. More specifically, a Reinforcement Learning approach using a linear combination of RBFs is assessed and compared to the classic NMPC coupled with EKF methodology in the Van de Vusse case study. The RFB-based RL is an interesting approach as an alternative to neural-network-based RL since it can guarantee convergence, while neural networks cannot. However, neural networks are often more used due to their higher capacity of generalization. A hyperbolic function is proposed as a gradual activator of the optimization term in the reward function. In addition, a sensitivity analysis on some of the training parameters is presented as guidelines of the actor-critic-RBF-based RL training approach. It was shown that both RL and NLMPC controllers can achieve the control and economic goals under unmeasured disturbance operation scenarios with only minor differences in performance. In the training sensitivity analysis, it was shown that: the increase in exploration must be accompanied by an increase in the number of episodes; the concentration maximization activation function cannot be excessively stiff; and there is an optimal value for the weight on the concentration maximization in the reward function. Finally, it was shown that linear combinations of RBFs are a successful alternative to neural networks.

Chapter 9

Future Work

The present work opened some research fronts for future works, here some are outlined:

- *More detailed dynamic approximations* that preserve the ability to converge to the same steady-state point of the nonlinear static model. For instance, a structure based on the Volterra series, Wiener series, nonlinear functions, neural networks, etc. could present a more accurate behavior, although the identification stage would be more difficult. A generalized structure can be presented as:

$$\mathbf{y}_k^h - \mathbf{y}^s(\mathbf{u}, \boldsymbol{\theta}) = \mathcal{F}(\mathbf{y}_{k-1}^h - \mathbf{y}^s(\mathbf{u}, \boldsymbol{\theta}), \mathbf{y}_{k-2}^h - \mathbf{y}^s(\mathbf{u}, \boldsymbol{\theta}), \dots) \quad (9.1)$$

in which any selection function $\mathcal{F}(\cdot)$ must converge uniformly to zero:

$$\lim_{k \rightarrow \infty} \mathcal{F}(\mathbf{y}_{k-1}^h - \mathbf{y}^s(\mathbf{u}, \boldsymbol{\theta}), \mathbf{y}_{k-2}^h - \mathbf{y}^s(\mathbf{u}, \boldsymbol{\theta}), \dots) = \mathbf{0} \quad (9.2)$$

therefore, the steady state of the approximate dynamic model will match the steady state of the rigorous static model:

$$\mathbf{y}_\infty^h = \mathbf{y}^s(\mathbf{u}, \boldsymbol{\theta}) \quad (9.3)$$

- *More advanced adaptation schemes.* As previously stressed, a more detailed dynamic approximation would result in a more difficult identification stage, where sufficiently-excited data should be available. A linear model would be enough for several applications, but recursive adaptation approaches proved to be not suitable due to the arising of undesirable oscillatory and unstable modes. Therefore, it would be interesting to have a separate adaptation layer in a different frequency of the control and optimization layer to guarantee an accurate dynamic approximation even when simple model

structures are used.

- *Systematic methodology for EKF tuning and initialization:* Although EKF was proposed in the 1960s and there have been several successful applications and advances in the technology, there is still a lack of systematic methodologies for tuning and initialization of the observer, which is mainly the cause why many practitioners avoid working with EKF. Not to mention that EKF is the simplest nonlinear Bayesian observer available, this situation is still more dramatic for more complex Bayesian estimators. Therefore, this is an important direction to enable real industrial applications of HRT0. There are some works in this direction, but they lack further testing and validation, such as the works of SALAU *et al.* (2009), SALAU *et al.* (2014), and SCHNEIDER and GEORGAKIS (2013).
- *Integration between HRT0 and SOC.* Different criteria can be proposed for removing input variables for each active constraint in the H-matrix reconfiguration algorithm. In addition, the methodology for calculating the H-matrix only takes the static model into account, therefore the performance during the transients is not optimal. Another way to calculate the H-matrix to take the dynamic or the approximate dynamics into account can be further investigated.
- *Different classes of dynamic behavior.* A broader investigation should be carried out to identify other limitations that may hinder some sorts of applications, for instance, dead time, inverse response in different time scales, inversion of the gain sign, bifurcations, oscillatory behavior, and the effect of disturbances are a matter of interest. It has been shown that unmodeled dead time could destabilize the HRT0 algorithm (CURVELO *et al.*, 2021), but no strategy to overcome this issue has been proposed yet.
- *Structural uncertainty.* This work has proposed the use of M_Ay-GP in the optimization and control layer. However, the actual proposition does not cope with a data reconciliation or dynamic observer stage. It is argued in the literature that problem adaptation algorithms do not need any data reconciliation or model adaptation to drive the plant toward its optimal point, which is true. However, for real applications, the data reconciliation or dynamic observers provide valuable information about the process and instrumentation performance. Therefore, a unified framework between problem adaptation and HRT0 would have greater applicability, but there is no clear description of how to proceed with this integration, especially with the use

of GP modifiers. Another interesting front would be how to bring transient information for the construction of the GP modifiers.

- *Plant-wide optimization.* The HRTO literature still lacks large-scale system case studies, all previous work, has focused only on small-scale benchmarks. The only real implementation was presented in the current work at a small-scale experimental rig. These kinds of studies would raise interesting insights and recommendations about possible future applications in real process plants.

Bibliography

- AHO, M., BERGMAN, S., HAMMARSTRÖM, L., YLI-OPAS, K., PELKOLA, A., SOURANDER, M., 2009, "Closed loop dynamic optimization of a petroleum refinery process", *IFAC Proceedings Volumes*, v. 42, n. 2, pp. 257–262.
- ALEXANDER, R., CAMPANI, G., DINH, S., LIMA, F. V., 2020, "Challenges and Opportunities on Nonlinear State Estimation of Chemical and Biochemical Processes", *Processes*, v. 8, n. 11. doi: <10.3390/pr8111462>.
- ALMEIDA NT, E., SECCHI, A., 2011, "Dynamic optimization of a FCC converter unit: numerical analysis", *Brazilian Journal of Chemical Engineering*, v. 28 (03), pp. 117–136. doi: <10.1590/S0104-66322011000100014>.
- ALSTAD, V., SKOGESTAD, S., 2007, "Null Space Method for Selecting Optimal Measurement Combinations as Controlled Variables", *Industrial & Engineering Chemistry Research*, v. 46, n. 3, pp. 846–853. doi: <10.1021/ie060285+>.
- ALSTAD, V., SKOGESTAD, S., HORI, E. S., 2009, "Optimal measurement combinations as controlled variables", *Journal of Process Control*, v. 19, n. 1, pp. 138–148. doi: <10.1016/j.jprocont.2008.01.002>.
- ANDERSON, C. W., HITTLE, D. C., KATZ, A. D., KRETCHMAR, R. M., 1997, "Synthesis of reinforcement learning, neural networks and PI control applied to a simulated heating coil", *Artificial Intelligence in Engineering*, v. 11, n. 4, pp. 421–429.
- ANDERSSON, J. A. E., GILLIS, J., HORN, G., RAWLINGS, J. B., DIEHL, M., 2019, "CasADi – A software framework for nonlinear optimization and optimal control", *Mathematical Programming Computation*, v. 11, n. 1, pp. 1–36. doi: <10.1007/s12532-018-0139-4>.
- ANDRYCHOWICZ, O. M., BAKER, B., CHOCIEJ, M., JÓZEFOWICZ, R., MCGREW, B., PACHOCKI, J., PETRON, A., PLAPPERT, M., POWELL, G.,

- RAY, A., SCHNEIDER, J., SIDOR, S., TOBIN, J., WELINDER, P., WENG, L., ZAREMBA, W., 2020, "Learning dexterous in-hand manipulation", *The International Journal of Robotics Research*, v. 39, n. 1, pp. 3–20.
- APIO, A., DAMBROS, J. W., DIEHL, F. C., FARENZENA, M., TRIERWEILER, J. O., 2019, "PDG Pressure Estimation in Offshore Oil Well: Extended Kalman Filter vs. Artificial Neural Networks", *IFAC-PapersOnLine*, v. 52, n. 1, pp. 508–513. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019.
- ARISTA, F., PORCEL, A., VILLAR, P., TENDRUP, T., 2006, "Real Time Optimization of ULSD Production". In: *ERTC Asset Maximization*, Paris.
- ATTA, K. T., GUAY, M., 2019, "Adaptive amplitude fast proportional integral phase extremum seeking control for a class of nonlinear system", *Journal of Process Control*, v. 83, pp. 147–154. doi: <10.1016/j.jprocont.2018.10.006>.
- BADGWELL, T. A., LEE, J. H., LIU, K. H., 2018, *Reinforcement Learning – Overview of Recent Progress and Implications for Process Control*, v. 44. Elsevier Masson SAS. ISBN: 9780444642417. Availability: <<https://doi.org/10.1016/B978-0-444-64241-7.50008-2>>.
- BAMBERGER, W., ISERMANN, R., 1978, "Adaptive on-line steady-state optimization of slow dynamic processes", *Automatica*, v. 14, n. 3, pp. 223–230.
- BARD, Y., 1970, "Comparison of Gradient Methods for the Solution of Nonlinear Parameter Estimation Problems", *SIAM J. Numer. Anal.*, v. 7, n. 1, pp. 157–186.
- BESL, H., KOSSMAN, W., CROWE, T. J., CARACOTSIOS, M., 1998, "Non-traditional optimization for isom unit improves profits", *Oil and Gas Journal*, v. 96, n. 19. Availability: <<http://worldcat.org/issn/00301388>>.
- BIEGLER, L. T., 1984a, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation", *Computers & Chemical Engineering*, v. 8, n. 3, pp. 243–247. doi: <[https://doi.org/10.1016/0098-1354\(84\)87012-X](https://doi.org/10.1016/0098-1354(84)87012-X)>.
- BIEGLER, L. T., 1984b, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation", *Computers & Chemical Engineering*, v. 8, pp. 243–247.

- BIEGLER, L. T., 2010, *Nonlinear Programming*. Society for Industrial and Applied Mathematics.
- BIEGLER, L. T., DONG LANG, Y., LIN, W., 2014, "Multi-scale optimization for process systems engineering", *Computers & Chemical Engineering*, v. 60, pp. 17 – 30. doi: <<https://doi.org/10.1016/j.compchemeng.2013.07.009>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0098135413002299>>.
- BIEGLER, L., GROSSMANN, I., WESTERBERG, A., 1985, "A note on approximation techniques used for process optimization", *Computers & Chemical Engineering*, v. 9, n. 2, pp. 201 – 206. doi: <[https://doi.org/10.1016/0098-1354\(85\)85010-9](https://doi.org/10.1016/0098-1354(85)85010-9)>. Availability: <<http://www.sciencedirect.com/science/article/pii/0098135485850109>>.
- BONVIN, D., 1998, "Optimal operation of batch reactors - A personal view", *Journal of Process Control*, v. 8, n. 5-6, pp. 355–368. doi: <[10.1016/S0959-1524\(98\)00010-9](https://doi.org/10.1016/S0959-1524(98)00010-9)>.
- BOX, G. E. P., 1957, "Evolutionary Operation: A Method for Increasing Industrial Productivity", *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, v. 6, n. 2, pp. 81–101.
- BOX, G. E. P., DRAPER, N. R., 1969, *Evolutionary operation: a statistical method for process improvement*. New York, John Wiley & Sons.
- BRADTKE, S. J., YDSTIE, B. E., BARTO, A. G., 1994, "Adaptive linear quadratic control using policy iteration". In: *American Control Conference, 1994*, v. 3, pp. 3475–3479. IEEE.
- BRDYŚ, M., CHEN, S., ROBERTS, P. D., 1986, "An extension to the modified two-step algorithm for steady-state system optimization and parameter estimation", *International Journal of Systems Science*, v. 17, n. 8, pp. 1229–1243. doi: <[10.1080/00207728608926883](https://doi.org/10.1080/00207728608926883)>.
- BRDYŚ, M., TATJEWSKI, P., 1994, "An Algorithm for Steady-State Optimizing Dual Control of Uncertain Plants", *IFAC Proceedings Volumes*, v. 27, n. 11, pp. 215–220. doi: <[10.1016/S1474-6670\(17\)47650-6](https://doi.org/10.1016/S1474-6670(17)47650-6)>.
- BRDYS, M. A., TATJEWSKI, P., 2005, *Iterative Algorithms for Multilayer Optimizing Control*, v. 1. Imperial College Press. ISBN: 978-1-86094-724-7.

- BROOKS, K., 2003, "Real Time Control and Optimisation – Current Status, New Developments and Future Possibilities". 12.
- BUNIN, G. A., 2014, "On the equivalence between the modifier-adaptation and trust-region frameworks", *Computers & Chemical Engineering*, v. 71, pp. 154 – 157. doi: <<https://doi.org/10.1016/j.compchemeng.2014.07.028>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0098135414002324>>.
- BUNIN, G. A., FRANÇOIS, G., BONVIN, D., 2013. "Sufficient Conditions for Feasibility and Optimality of Real-Time Optimization Schemes - I. Theoretical Foundations". .
- BUSONI, L., BABUSKA, R., DE SCHUTTER, B., ERNST, D., 2010, *Reinforcement learning and dynamic programming using function approximators*, v. 39. Boca Raton, CRC press.
- CADAVID, J., HERNÁNDEZ, R., ENGELL, S., 2017, "Speed-up of Iterative Real-Time Optimization by Estimating the Steady States in the Transient Phase using Nonlinear System Identification", *IFAC-PapersOnLine*, v. 50, n. 1, pp. 11269 – 11274. 20th IFAC World Congress.
- CÂMARA, M. M., QUELHAS, A. D., PINTO, J. C., 2016, "Performance evaluation of real industrial RTO systems", *Processes*, v. 4, n. 4, pp. 1–20. doi: <[10.3390/pr4040044](https://doi.org/10.3390/pr4040044)>.
- CAMOLESI, V. J., MORO, L. L. F., ZANIN, A. C., 2008, "Implantação de um otimizador em tempo real (RTO) no conversor de uma unidade de craqueamento catalítico", *Sba Controle & Automação Sociedade Brasileira de Automatica*, v. 19 (06), pp. 128 – 137.
- CAMPOS, M., TEIXEIRA, H., LIPORACE, F., GOMES, M., 2009, "Challenges and problems with advanced control and optimization technologies", *IFAC Proceedings Volumes*, v. 42, n. 11, pp. 1–8. doi: <<https://doi.org/10.3182/20090712-4-TR-2008.00003>>. 7th IFAC Symposium on Advanced Control of Chemical Processes.
- CAMPOS, M., GOMES, M., SOUZA, A., BARROS, A., 2012, "Optimisation of natural gas plant - Gains in profitability, stability and energy efficiency", *International Gas Union World Gas Conference Papers*, v. 3 (01), pp. 2089–2120.

- CAO, Y., 2004, "Constrained Self-Optimizing Control via Differentiation1", *IFAC Proceedings Volumes*, v. 37, n. 1, pp. 63–70. doi: <[https://doi.org/10.1016/S1474-6670\(17\)38710-4](https://doi.org/10.1016/S1474-6670(17)38710-4)>. Availability: <<https://www.sciencedirect.com/science/article/pii/S1474667017387104>>. 7th International Symposium on Advanced Control of Chemical Processes (ADCHEM 2003), Hong-Kong, 11-14 January 2004.
- CAO, Y., KARIWALA, V., 2008, "Bidirectional branch and bound for controlled variable selection: Part I. Principles and minimum singular value criterion", *Computers & Chemical Engineering*, v. 32, n. 10, pp. 2306–2319. doi: <<https://doi.org/10.1016/j.compchemeng.2007.11.011>>. Availability: <<https://www.sciencedirect.com/science/article/pii/S0098135407002906>>.
- CASSOL, G. O., CAMPOS, G. V., THOMAZ, D. M., CAPRON, B. D., SECCHI, A. R., 2018a, *Reinforcement Learning Applied to Process Control: A Van der Vusse Reactor Case Study*, v. 44. Elsevier Masson SAS. ISBN: 9780444642417. Availability: <<https://doi.org/10.1016/B978-0-444-64241-7.50087-2>>.
- CASSOL, G., CAMPOS, G., THOMAZ, D., CAPRON, B., SECCHI, A., 2018b, "Reinforcement Learning Applied to Process Control: A Van der Vusse Reactor Case Study". In: *13th International Symposium on Process Systems Engineering (PSE 2018)*, v. 44, *Computer Aided Chemical Engineering*, pp. 553–558, b.
- CHACHUAT, B., MARCHETTI, A., BONVIN, D., 2008a, "Process optimization via constraints adaptation", *Journal of Process Control*, v. 18, n. 3, pp. 244 – 257. doi: <<https://doi.org/10.1016/j.jprocont.2007.07.001>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0959152407001138>>. Festschrift honouring Professor Dale Seborg.
- CHACHUAT, B., STRINIVANSAN, B., BONVIN, D., 2009, "Adaption strategies for real-time optimization", *Computers and Chemical Engineering*, v. 33, pp. 1557–1567.
- CHACHUAT, B., SRINIVASAN, B., BONVIN, D., 2008b, "Model parameterization tailored to real-time optimization". In: Braunschweig, B., Joulia, X. (Eds.), *18th European Symposium on Computer Aided Process Engineering*, v. 25, *Computer Aided Chemical Engineering*, Elsevier, pp. 1 – 13, b.

- CHEN, C. Y., JOSEPH, B., 1987, "On-Line Optimization Using a Two-Phase Approach: An Application Study", *Industrial and Engineering Chemistry Research*, v. 26, n. 9, pp. 1924–1930. doi: <10.1021/ie00069a034>.
- COSTELLO, S., FRANÇOIS, G., BONVIN, D., 2015, "Directional real-time optimization applied to a kite-control simulation benchmark". In: *2015 European Control Conference (ECC)*, pp. 1594–1601, July.
- COSTELLO, S., FRANÇOIS, G., BONVIN, D., 2018, "Real-Time Optimizing Control of an Experimental Crosswind Power Kite", *IEEE Transactions on Control Systems Technology*, v. 26, n. 2, pp. 507–522.
- COSTELLO, S., FRANÇOIS, G., BONVIN, D., 2016, "A Directional Modifier-Adaptation Algorithm for Real-Time Optimization", *Journal of Process Control*, v. 39, pp. 64 – 76. doi: <https://doi.org/10.1016/j.jprocont.2015.11.008>. Availability: <http://www.sciencedirect.com/science/article/pii/S0959152415002395>.
- CURVELO, R., DE A. DELOU, P., DE SOUZA, M. B., SECCHI, A. R., 2021, "Investigation of the use of transient process data for steady-state Real-Time Optimization in presence of complex dynamics". In: Türkay, M., Gani, R. (Eds.), *31st European Symposium on Computer Aided Process Engineering*, v. 50, *Computer Aided Chemical Engineering*, Elsevier, pp. 1299–1305.
- DARBY, M. L., NIKOLAOU, M., JONES, J., NICHOLSON, D., 2011, "RTO: An overview and assessment of current practice", *Journal of Process Control*, v. 21, n. 6, pp. 874 – 884. doi: <https://doi.org/10.1016/j.jprocont.2011.03.009>.
- DE AVILA FERREIRA, T., WUILLEMIN, Z., MARCHETTI, A., BONVIN, D., 2019a, "Fast RTO Applied to a Commercial SOFC System", *IFAC-PapersOnLine*, v. 52, n. 1, pp. 40 – 45. doi: <https://doi.org/10.1016/j.ifacol.2019.06.034>. Availability: <http://www.sciencedirect.com/science/article/pii/S2405896319301193>. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019.
- DE AVILA FERREIRA, T., WUILLEMIN, Z., MARCHETTI, A., SALZMANN, C., HERLE, J. V., BONVIN, D., 2019b, "Real-time optimization of an experimental solid-oxide fuel-cell system", *Journal of Power Sources*, v. 429,

pp. 168 – 179. doi: <<https://doi.org/10.1016/j.jpowsour.2019.03.025>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0378775319302642>>.

DE AVILA FERREIRA, T., WUILLEMIN, Z., FAULWASSER, T., SALZMANN, C., HERLE, J. V., BONVIN, D., 2019c, “Enforcing optimal operation in solid-oxide fuel-cell systems”, *Energy*, v. 181, pp. 281 – 293. doi: <<https://doi.org/10.1016/j.energy.2019.04.188>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0360544219308278>>.

DE AVILA FERREIRA, T., FRANÇOIS, G., MARCHETTI, A. G., BONVIN, D., 2017, “Use of Transient Measurements for Static Real-Time Optimization”, *IFAC-PapersOnLine*, v. 50, n. 1, pp. 5737 – 5742. 20th IFAC World Congress.

DE AVILA FERREIRA, T., SHUKLA, H. A., FAULWASSER, T., JONES, C. N., BONVIN, D., 2018, “Real-Time optimization of Uncertain Process Systems via Modifier Adaptation and Gaussian Processes”. In: *2018 European Control Conference (ECC)*, pp. 465–470, June.

DE GOUVÊA, M. T., ODLOAK, D., 1998, “One-layer real time optimization of LPG production in the FCC unit: procedure, advantages and disadvantages”, *Computers & Chemical Engineering*, v. 22, pp. S191–S198. doi: <[https://doi.org/10.1016/S0098-1354\(98\)00054-4](https://doi.org/10.1016/S0098-1354(98)00054-4)>. Availability: <<https://www.sciencedirect.com/science/article/pii/S0098135498000544>>. European Symposium on Computer Aided Process Engineering-8.

DE MENEZES, D., PRATA, D., SECCHI, A., PINTO, J., 2021, “A review on robust M-estimators for regression analysis”, *Computers & Chemical Engineering*, v. 147, pp. 107254. doi: <[10.1016/j.compchemeng.2021.107254](https://doi.org/10.1016/j.compchemeng.2021.107254)>.

DE SOUZA, G., ODLOAK, D., ZANIN, A. C., 2010, “Real time optimization (RTO) with model predictive control (MPC)”, *Computers & Chemical Engineering*, v. 34, n. 12, pp. 1999–2006. doi: <<https://doi.org/10.1016/j.compchemeng.2010.07.001>>. 10th International Symposium on Process Systems Engineering, Salvador, Bahia, Brasil, 16-20 August 2009.

DEL RIO CHANONA, E. A., PETSAGKOURAKIS, P., BRADFORD, E., GRACIANO, J. E. A., CHACHUAT, B., 2021, “Real-time optimization meets

Bayesian optimization and derivative-free optimization: A tale of modifier adaptation”, *Computers & Chemical Engineering*, v. 147, pp. 107249. doi: <10.1016/j.compchemeng.2021.107249>.

DEL RIO CHANONA, E., GRACIANO, J. A., BRADFORD, E., CHACHUAT, B., 2019, “Modifier-Adaptation Schemes Employing Gaussian Processes and Trust Regions for Real-Time Optimization**The first two authors contributed equally to the paper”, *IFAC-PapersOnLine*, v. 52, n. 1, pp. 52 – 57. doi: <<https://doi.org/10.1016/j.ifacol.2019.06.036>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S2405896319301211>>. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019.

DELOU, P. A., BERNARDINO, L. F., CAPRON, B. D., SECCHI, A. R., 2021a, *A Comparison between Process Control Strategies: Reinforcement Learning with RBF and NMPC coupled with EKF*. Anais eletrônicos.

DELOU, P., CURVELO, R., DE SOUZA, M. B., SECCHI, A. R., 2021b, “Development of Hybrid RTO approaches in the absence of a rigorous dynamic model by the use of Hammerstein model structures”. In: Türkay, M., Gani, R. (Eds.), *31st European Symposium on Computer Aided Process Engineering*, v. 50, *Computer Aided Chemical Engineering*, Elsevier, pp. 259–265, b. Availability: <<https://www.sciencedirect.com/science/article/pii/B9780323885065500425>>.

DELOU, P., CURVELO, R., DE SOUZA, M. B., SECCHI, A. R., 2021c, “Steady-state real-time optimization using transient measurements in the absence of a dynamic mechanistic model: A framework of HRTM integrated with Adaptive Self-Optimizing IHMPC”, *Journal of Process Control*, v. 106, pp. 1–19.

DELOU, P. A., RIBEIRO, L. D., PAIVA, C. R., NIEDERBERGER, J., GOMES, M. V. C., SECCHI, A. R., 2021d, “A Real-Time Optimization Strategy for Small-Scale Facilities and Implementation in a Gas Processing Unit”, *Processes*, v. 9, n. 7. doi: <10.3390/pr9071179>.

DELOU, P. D. A., BERNARDINO, L. F., CAPRON, B. D. O., SECCHI, A. R., 2023a, “A comparison between process control strategies: reinforcement learning with RBFs and NMPC coupled with EKF”, *Brazilian Journal of Chemical Engineering*. doi: <10.1007/s43153-023-00351-w>.

- DELOU, P. D. A., MATIAS, J., JÄSCHKE, J., DE SOUZA, M. B., SECCHI, A. R., 2023b, "Steady-state real-time optimization using transient measurements and approximated Hammerstein dynamic model: A proof of concept in an experimental rig", *Journal of Process Control*, v. 132, pp. 103111. doi: <<https://doi.org/10.1016/j.jprocont.2023.103111>>.
- DEMUNER, R. B., DE AZEVEDO DELOU, P., SECCHI, A. R., 2022, "Tracking necessary condition of optimality by a data-driven solution combining steady-state and transient data", *Journal of Process Control*, v. 118, pp. 37–54. doi: <<https://doi.org/10.1016/j.jprocont.2022.08.001>>.
- DI MARCO, R., SEMINO, D., BRAMBILLA, A., 1997, "From Linear to Nonlinear Model Predictive Control: Comparison of Different Algorithms", *Industrial & Engineering Chemistry Research*, v. 36, n. 5, pp. 1708–1716. doi: <[10.1021/ie960552j](https://doi.org/10.1021/ie960552j)>.
- DOCHAIN, D., PERRIER, M., GUAY, M., 2011, "Extremum seeking control and its application to process and reaction systems: A survey", *Mathematics and Computers in Simulation*, v. 82, n. 3, pp. 369–380. 6th Vienna International Conference on Mathematical Modelling.
- ELLIS, J. E., KAMBHAMPATI, C., SHENG, G., ROBERTS, P. D., 1988, "Approaches to the optimizing control problem", *International Journal of Systems Science*, v. 19, n. 10, pp. 1969–1985. doi: <[10.1080/00207728808964092](https://doi.org/10.1080/00207728808964092)>.
- ELLIS, M., DURAND, H., CHRISTOFIDES, P. D., 2014, "A tutorial review of economic model predictive control methods", *Journal of Process Control*, v. 24, n. 8, pp. 1156–1178. doi: <<https://doi.org/10.1016/j.jprocont.2014.03.010>>. Economic nonlinear model predictive control.
- ELLIS, M., LIU, J., CHRISTOFIDES, P. D., 2017, "Brief Overview of EMPC Methods and Some Preliminary Results". In: *Economic Model Predictive Control: Theory, Formulations and Chemical Process Applications*, pp. 57–73, Cham, Springer International Publishing. ISBN: 978-3-319-41108-8.
- ENGELL, S., 2007, "Feedback control for optimal process operation", *Journal of Process Control*, v. 17, n. 3, pp. 203–219. doi: <[10.1016/j.jprocont.2006.10.011](https://doi.org/10.1016/j.jprocont.2006.10.011)>. Special Issue ADCHEM 2006 Symposium.

- FARIA, R., CAPRON, B., SECCHI, A., DE SOUZA, M., 2024, "A data-driven tracking control framework using physics-informed neural networks and deep reinforcement learning for dynamical systems", *Engineering Applications of Artificial Intelligence*, v. 127, pp. 107256. doi: <<https://doi.org/10.1016/j.engappai.2023.107256>>.
- FARIA, R. D. R., CAPRON, B. D. O., SECCHI, A. R., DE SOUZA JR., M. B., 2022, "Where Reinforcement Learning Meets Process Control: Review and Guidelines", *Processes*, v. 10, n. 11, pp. 2311. doi: <10.3390/pr10112311>.
- FARIA, R. D. R., CAPRON, B. D. O., DE SOUZA JR., M. B., SECCHI, A. R., 2023, "One-Layer Real-Time Optimization Using Reinforcement Learning: A Review with Guidelines", *Processes*, v. 11, n. 1. doi: <10.3390/pr11010123>.
- FAULWASSER, T., BONVIN, D., 2014, "On the Use of Second-Order Modifiers for Real-Time Optimization", *IFAC Proceedings Volumes*, v. 47, n. 3, pp. 7622 – 7628. doi: <<https://doi.org/10.3182/20140824-6-ZA-1003.00735>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S1474667016428144>>. 19th IFAC World Congress.
- FLEURY, C., 1989, "First and second order convex approximation strategies in structural optimization." *Structural Optimization*, v. 1, n. 1, pp. 3–10. doi: <<https://doi.org/10.1007/BF01743804>>.
- FORBES, J. F., MARLIN, T. E., 1994, "Model Accuracy for Economic Optimizing Controllers: The Bias Update Case", *Industrial & Engineering Chemistry Research*, v. 33, n. 8, pp. 1919–1929. doi: <10.1021/ie00032a006>.
- FORBES, J., MARLIN, T., 1996, "Design cost: a systematic approach to technology selection for model-based real-time optimization systems", *Computers & Chemical Engineering*, v. 20, n. 6, pp. 717 – 734. doi: <[https://doi.org/10.1016/0098-1354\(95\)00205-7](https://doi.org/10.1016/0098-1354(95)00205-7)>. Availability: <<http://www.sciencedirect.com/science/article/pii/0098135495002057>>. Fifth International Symposium on Process Systems Engineering.
- FORBES, J., MARLIN, T., MACGREGOR, J., 1994, "Model adequacy requirements for optimizing plant operations", *Computers & Chemical Engineering*, v. 18, n. 6, pp. 497 – 510. doi: <[https://doi.org/10.1016/0098-1354\(94\)00032-8](https://doi.org/10.1016/0098-1354(94)00032-8)>.

[//doi.org/10.1016/0098-1354\(93\)E0005-T](https://doi.org/10.1016/0098-1354(93)E0005-T). Availability: <http://www.sciencedirect.com/science/article/pii/0098135493E0005T>. *An International Journal of Computer Applications in Chemical Engineering*.

FRANÇOIS, G., SRINIVASAN, B., BONVIN, D., 2005, "Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty", *Journal of Process Control*, v. 15, n. 6, pp. 701–712. doi: <10.1016/j.jprocont.2004.11.006>.

FRANÇOIS, G., SRINIVASAN, B., BONVIN, D., 2012, "Comparison of six implicit real-time optimization schemes", *Journal European des Systemes Automatises*, v. 46, n. 2-3, pp. 291–305. doi: <10.3166/JESA.46.291-305>.

FRANCOIS, G., BONVIN, D., 2013, "Chapter One - Measurement-Based Real-Time Optimization of Chemical Processes". In: Pushpavanam, S. (Ed.), *Control and Optimisation of Process Systems*, v. 43, *Advances in Chemical Engineering*, Academic Press, pp. 1 – 50. Availability: <http://www.sciencedirect.com/science/article/pii/B9780123965240000015>.

FRANÇOIS, G., BONVIN, D., 2013, "Use of Convex Model Approximations for Real-Time Optimization via Modifier Adaptation", *Industrial & Engineering Chemistry Research*, v. 52, n. 33, pp. 11614–11625. doi: <10.1021/ie3032372>.

FRANÇOIS, G., BONVIN, D., 2014, "Use of Transient Measurements for the Optimization of Steady-State Performance via Modifier Adaptation", *Industrial & Engineering Chemistry Research*, v. 53, n. 13, pp. 5148–5159. doi: <10.1021/ie401392s>.

FRIEDMAN, Y., 2005, "Has the advanced process control industry completely collapsed?" *Hydrocarbon Processing*, v. 84 (01), pp. 15–16.

FRIEDMAN, Y. Z., 1995, "What's wrong with unit closed loop optimization?" *Hydrocarbon Processing*, v. 74, n. 10 (10). Availability: <https://www.osti.gov/biblio/122684>.

FRIEDMAN, Y. Z., 1998, "More about closed-loop optimization", *Hydrocarbon Processing*, (08).

- GAO, W., ENGELL, S., 2005, "Iterative set-point optimization of batch chromatography", *Computers & Chemical Engineering*, v. 29, n. 6, pp. 1401 – 1409. doi: <<https://doi.org/10.1016/j.compchemeng.2005.02.035>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0098135405000281>>. Selected Papers Presented at the 14th European Symposium on Computer Aided Process Engineering.
- GAO, W., ENGELL, S., 2016, "Using Transient Measurements in Iterative Steady-State Optimizing Control". In: Kravanja, Z., Bogataj, M. (Eds.), *26th European Symposium on Computer Aided Process Engineering*, v. 38, *Computer Aided Chemical Engineering*, Elsevier, pp. 511 – 516. Availability: <<http://www.sciencedirect.com/science/article/pii/B9780444634283500904>>.
- GAO, W., HERNÁNDEZ, R., ENGELL, S., 2016a, "A Study of Explorative Moves during Modifier Adaptation with Quadratic Approximation", *Processes*, v. 4, n. 4, pp. 45. doi: <<https://doi.org/10.3390/pr4040045>>.
- GAO, W., WENZEL, S., ENGELL, S., 2016b, "A reliable modifier-adaptation strategy for real-time optimization", *Computers & Chemical Engineering*, v. 91, pp. 318 – 328. doi: <<https://doi.org/10.1016/j.compchemeng.2016.03.019>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0098135416300813>>. 12th International Symposium on Process Systems Engineering & 25th European Symposium of Computer Aided Process Engineering (PSE-2015/ESCAPE-25), 31 May - 4 June 2015, Copenhagen, Denmark.
- GIRALDO, S. A., MELO, P. A., SECCHI, A. R., 2019, "Tuning of Model Predictive Control Based on Hybrid Optimization", *IFAC-PapersOnLine*, v. 52, n. 1, pp. 136–141. doi: <[10.1016/j.ifacol.2019.06.050](https://doi.org/10.1016/j.ifacol.2019.06.050)>.
- GOLDEN, M. P., YDSTIE, B. E., 1989, "Adaptive extremum control using approximate process models", *AIChE Journal*, v. 35, n. 7, pp. 1157–1169. doi: <[10.1002/aic.690350711](https://doi.org/10.1002/aic.690350711)>. Availability: <<https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690350711>>.
- GOMES, M. V. C., 2007, *Otimização seqüencial por aproximações – uma aplicação em tempo real para o refino de petróleo*. Phd thesis, Universidade Federal do

Rio de Janeiro, COPPE, Rio de Janeiro, RJ, August. Available at <https://rb.gy/l2y4vq>.

- GONZÁLEZ, A. H., ODLOAK, D., 2009, "A stable MPC with zone control", *Journal of Process Control*, v. 19, n. 1, pp. 110–122. doi: <<https://doi.org/10.1016/j.jprocont.2008.01.003>>.
- GRACIANO, J. E. A., JÄSCHKE, J., LE ROUX, G. A., BIEGLER, L. T., 2015, "Integrating self-optimizing control and real-time optimization using zone control MPC", *Journal of Process Control*, v. 34, pp. 35–48. doi: <[10.1016/j.jprocont.2015.07.003](https://doi.org/10.1016/j.jprocont.2015.07.003)>.
- GROS, S., SRINIVASAN, B., BONVIN, D., 2006, "Robust predictive control based on neighboring extremals", *Journal of Process Control*, v. 16, n. 3, pp. 243–253. doi: <[10.1016/j.jprocont.2005.06.005](https://doi.org/10.1016/j.jprocont.2005.06.005)>. Selected Papers from Dycops 7 (2004), Cambridge, Massachusetts.
- GROS, S., SRINIVASAN, B., BONVIN, D., 2009, "Optimizing control based on output feedback", *Computers and Chemical Engineering*, v. 33, n. 1, pp. 191–198. doi: <[10.1016/j.compchemeng.2008.07.017](https://doi.org/10.1016/j.compchemeng.2008.07.017)>.
- GUAY, M., ZHANG, T., 2003, "Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties", *Automatica*, v. 39, n. 7, pp. 1283–1293. doi: <[10.1016/S0005-1098\(03\)00105-5](https://doi.org/10.1016/S0005-1098(03)00105-5)>.
- HASELTINE, E., RAWLINGS, J., 2005, "Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation", *Industrial and Engineering Chemistry Research*, v. 44 (04), pp. 2451–2460. doi: <[10.1021/ie0343081](https://doi.org/10.1021/ie0343081)>.
- HEESS, N., TB, D., SRIRAM, S., LEMMON, J., MEREL, J., WAYNE, G., TASSA, Y., EREZ, T., WANG, Z., ESLAMI, A., RIEDMILLER, M., SILVER, D., 2017, "Emergence of Locomotion Behaviours in Rich Environments", (07).
- HERNANDEZ, R., DREIMANN, J., ENGELL, S., 2018a, "Reliable Iterative RTO of a Continuously Operated Hydroformylation Process", *IFAC-PapersOnLine*, v. 51, n. 18, pp. 61 – 66. doi: <<https://doi.org/10.1016/j.ifacol.2018.09.249>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S2405896318319256>>. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

- HERNANDEZ, R., DREIMANN, J., VORHOLT, A., BEHR, A., ENGELL, S., 2018b, "Iterative Real-Time Optimization Scheme for Optimal Operation of Chemical Processes under Uncertainty: Proof of Concept in a Miniplant", *Industrial & Engineering Chemistry Research*, v. 57, n. 26, pp. 8750–8770. doi: <10.1021/acs.iecr.8b00615>.
- HERNÁNDEZ-DEL OLMO, F., GAUDIOSO, E., DORMIDO, R., DURO, N., 2017, "Tackling the Start-Up of a Reinforcement Learning Agent for the Control of Wastewater Treatment Plants", *Knowledge-Based Systems*.
- HERNÁNDEZ, R., BUCKOVÁ, M., ENGELL, S., 2017, "An efficient RTO scheme for the optimal operation of chemical processes under uncertainty". In: *2017 21st International Conference on Process Control (PC)*, pp. 364–369, June.
- HERNÁNDEZ, R., ENGELL, S., 2017, "Stochastic Approximation in On-line Steady State Optimization Under Noisy Measurements". In: Espuña, A., Graells, M., Puigjaner, L. (Eds.), *27th European Symposium on Computer Aided Process Engineering*, v. 40, *Computer Aided Chemical Engineering*, Elsevier, pp. 1747 – 1752. Availability: <<http://www.sciencedirect.com/science/article/pii/B9780444639653502932>>.
- HINDMARSH, A. C., BROWN, P. N., GRANT, K. E., LEE, S. L., SERBAN, R., SHUMAKER, D. E., WOODWARD, C. S., 2005, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers", *ACM Transactions on Mathematical Software (TOMS)*, v. 31, n. 3, pp. 363–396.
- HOFFMAN, M. D., BROCHU, E., DE FREITAS, N., 2010, "Portfolio Allocation for Bayesian Optimization", *ArXiv*, v. abs/1009.5419. Availability: <<https://api.semanticscholar.org/CorpusID:8053165>>.
- HOSKINS, J., HIMMELBLAU, D., 1992, "Process control via artificial neural networks and reinforcement learning", *Computers & chemical engineering*, v. 16, n. 4, pp. 241–251.
- HU, W., UMAR, L. M., XIAO, G., KARIWALA, V., 2012, "Local self-optimizing control of constrained processes", *Journal of Process Control*, v. 22, n. 2, pp. 488–493. doi: <<https://doi.org/10.1016/j.jprocont.2011.11.003>>. Availability: <<https://www.sciencedirect.com/science/article/pii/S0959152411002228>>.

- HUANG, J., LI, H., GUO, J., 2011, "Application of PSO in the improved real-time evolution". In: *2011 11th International Conference on Hybrid Intelligent Systems (HIS)*, pp. 318–323. IEEE.
- HUANG, R., 2010, *Nonlinear model predictive control and dynamic real time optimization for large-scale processes*. Ph.D. Thesis, Carnegie Mellon University.
- JANG, S., JOSEPH, B., MUKAI, H., 1987, "On-line optimization of constrained multivariable chemical processes", *AIChE Journal*, v. 33, n. 1, pp. 26–35. doi: <10.1002/aic.690330105>.
- JÄSCHKE, J., SKOGESTAD, S., 2011, "NCO tracking and self-optimizing control in the context of real-time optimization", *Journal of Process Control*, v. 21, n. 10, pp. 1407–1416. doi: <10.1016/j.jprocont.2011.07.001>.
- JÄSCHKE, J., CAO, Y., KARIWALA, V., 2017, "Self-optimizing control – A survey", *Annual Reviews in Control*, v. 43, pp. 199–223. doi: <10.1016/j.arcontrol.2017.03.001>.
- JOHANSEN, T. A., FOSS, B. A., 1992, "Nonlinear Local Model Representation For Adaptive Systems". In: *Singapore International Conference on Intelligent Control and Instrumentation [Proceedings 1992]*, v. 2, pp. 677–682, Feb.
- JOUIN, M., GOURIVEAU, R., HISSEL, D., PÉRA, M.-C., ZERHOUNI, N., 2016, "Particle filter-based prognostics: Review, discussion and perspectives", *Mechanical Systems and Signal Processing*, v. 72-73, pp. 2–31. doi: <10.1016/j.ymsp.2015.11.008>.
- JÄSCHKE, J., SKOGESTAD, S., 2010, "Self-optimizing Control and NCO tracking in the Context of Real-Time Optimization", *IFAC Proceedings Volumes*, v. 43, n. 5, pp. 607–612. doi: <10.3182/20100705-3-BE-2011.00101>. 9th IFAC Symposium on Dynamics and Control of Process Systems.
- KALLENBERGER, C., HAMEDOVIĆ, H., ZOUBIR, A. M., 2007, "Comparison of the Extended Kalman Filter and the unscented Kalman filter for parameter estimation in combustion engines". In: *2007 15th European Signal Processing Conference*, pp. 2424–2428.
- KARIWALA, V., CAO, Y., 2009, "Bidirectional branch and bound for controlled variable selection. Part II: Exact local method for self-optimizing control", *Computers & Chemical Engineering*, v. 33, n. 8, pp. 1402–1412. doi: <https://doi.org/10.1016/j.compchemeng.2009.>

01.014>. Availability: <<https://www.sciencedirect.com/science/article/pii/S0098135409000350>>.

KARIWALA, V., CAO, Y., 2010, "Bidirectional Branch and Bound for Controlled Variable Selection Part III: Local Average Loss Minimization", *IEEE Transactions on Industrial Informatics*, v. 6, n. 1, pp. 54–61. doi: <10.1109/TII.2009.2037494>.

KELLY, J. D., HEDENGREN, J. D., 2013, "A steady-state detection (SSD) algorithm to detect non-stationary drifts in processes", *Journal of Process Control*, v. 23, n. 3, pp. 326–331. doi: <<https://doi.org/10.1016/j.jprocont.2012.12.001>>. Availability: <<https://www.sciencedirect.com/science/article/pii/S0959152412002661>>.

KIM, H., JORDAN, M., SASTRY, S., NG, A., 2003, "Autonomous Helicopter Flight via Reinforcement Learning". In: Thrun, S., Saul, L., Schölkopf, B. (Eds.), *Advances in Neural Information Processing Systems*, v. 16, Cambridge. MIT Press.

KLATT, K.-U., ENGELL, S., 1998, "Gain-scheduling trajectory control of a continuous stirred tank reactor", *Computers & Chemical Engineering*, v. 22, n. 4, pp. 491–502.

KORYAKOVSKIY, I., KUDRUSS, M., BABUŠKA, R., CAARLS, W., KIRCHES, C., MOMBAUR, K., SCHLÖDER, J. P., VALLERY, H., 2017, "Benchmarking model-free and model-based optimal control", *Robotics and Autonomous Systems*, v. 92, pp. 81–90.

KRISHNAMOORTHY, D., SKOGESTAD, S., 2020, "Systematic design of active constraint switching using selectors", *Computers & Chemical Engineering*, v. 143, pp. 107106. doi: <10.1016/j.compchemeng.2020.107106>.

KRISHNAMOORTHY, D., SKOGESTAD, S., 2022, "Real-Time optimization as a feedback control problem – A review", *Computers & Chemical Engineering*, v. 161, pp. 107723.

KRISHNAMOORTHY, D., FOSS, B., SKOGESTAD, S., 2018a, "Steady-state real-time optimization using transient measurements", *Computers & Chemical Engineering*, v. 115, pp. 34 – 45. doi: <10.1016/j.compchemeng.2018.03.021>.

- KRISHNAMOORTHY, D., FOSS, B., SKOGESTAD, S., 2018b, "Steady-state real-time optimization using transient measurements", *Computers & Chemical Engineering*, v. 115, pp. 34 – 45. doi: <<https://doi.org/10.1016/j.compchemeng.2018.03.021>>.
- KRISHNAMOORTHY, D., JAHANSHAH, E., SKOGESTAD, S., 2019, "Feedback Real-Time Optimization Strategy Using a Novel Steady-state Gradient Estimate and Transient Measurements", *Industrial and Engineering Chemistry Research*, v. 58, n. 1, pp. 207–216. doi: <10.1021/acs.iecr.8b03137>.
- KRSTIĆ, M., 2000, "Performance improvement and limitations in extremum seeking control", *Systems and Control Letters*, v. 39, n. 5, pp. 313–326. doi: <10.1016/S0167-6911(99)00111-5>.
- KRSTIĆ, M., WANG, H. H., 2000, "Stability of extremum seeking feedback for general nonlinear dynamic systems", *Automatica*, v. 36, n. 4, pp. 595–601. doi: <10.1016/S0005-1098(99)00183-1>.
- LAVIOLA, J. J., 2003, "A comparison of unscented and Extended Kalman Filtering for estimating quaternion motion". In: *Proceedings of the 2003 American Control Conference, 2003.*, v. 3, pp. 2435–2440 vol.3.
- LEE, K. S., LEE, W.-K., 1985, "On-line optimizing control of a nonadiabatic fixed bed reactor", *AIChE Journal*, v. 31, n. 4, pp. 667–675. doi: <10.1002/aic.690310417>. Availability: <<https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690310417>>.
- LI, D., QIAN, L., JIN, Q., TAN, T., 2011, "Reinforcement learning control with adaptive gain for a *Saccharomyces cerevisiae* fermentation process", *Applied Soft Computing*, v. 11, n. 8, pp. 4488–4495.
- LIPORACE, F. S., GOMES, M. V., KATATA, A. C., ZANIN, A. C., MORO, L. F., PORFÍRIO, C. R., 2009, "PETROBRAS Experience Implementing Real Time Optimization". v. 27, *Computer Aided Chemical Engineering*, Elsevier, pp. 1245 – 1250.
- MANUM, H., SKOGESTAD, S., 2012, "Self-optimizing control with active set changes", *Journal of Process Control*, v. 22, n. 5, pp. 873–883. doi: <10.1016/j.jprocont.2012.02.015>.

- MARCHETTI, A., CHACHUAT, B., BONVIN, D., 2007, "REAL-TIME OPTIMIZATION OF CONTINUOUS PROCESSES VIA CONSTRAINTS ADAPTATION", *IFAC Proceedings Volumes*, v. 40, n. 5, pp. 45 – 50. doi: <<https://doi.org/10.3182/20070606-3-MX-2915.00006>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S1474667015316505>>. 8th IFAC Symposium on Dynamics and Control of Process Systems.
- MARCHETTI, A., CHACHUAT, B., BONVIN, D., 2009, "Modifier-adaptation methodology for real-time optimization", *Industrial and Engineering Chemistry Research*, v. 48, n. 13, pp. 6022–6033. doi: <10.1021/ie801352x>.
- MARCHETTI, A., CHACHUAT, B., BONVIN, D., 2010, "A dual modifier-adaptation approach for real-time optimization", *Journal of Process Control*, v. 20, n. 9, pp. 1027 – 1037. doi: <<https://doi.org/10.1016/j.jprocont.2010.06.006>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0959152410001174>>. ADCHEM 2009 Special Issue.
- MARCHETTI, A., GOPALAKRISHNAN, A., CHACHUAT, B., BONVIN, D., TSIKONIS, L., NAKAJO, A., WUILLEMIN, Z., VAN HERLE, J., 2011, "Robust Real-Time Optimization of a Solid Oxide Fuel Cell Stack", *Journal of Fuel Cell Science and Technology*, v. 8, n. 5 (06). doi: <10.1115/1.4003976>. Availability: <<https://doi.org/10.1115/1.4003976>>. 051001.
- MARCHETTI, A. G., DE AVILA FERREIRA, T., COSTELLO, S., BONVIN, D., 2020, "Modifier Adaptation as a Feedback Control Scheme", *Industrial & Engineering Chemistry Research*, v. 59, n. 6, pp. 2261–2274. doi: <10.1021/acs.iecr.9b04501>.
- MARCHETTI, A., 2013, "A new dual modifier-adaptation approach for iterative process optimization with inaccurate models", *Computers & Chemical Engineering*, v. 59, pp. 89 – 100. doi: <<https://doi.org/10.1016/j.compchemeng.2013.03.019>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0098135413000847>>. Selected papers from ESCAPE-22 (European Symposium on Computer Aided Process Engineering - 22), 17-20 June 2012, London, UK.

- MARCHETTI, A., CHACHUAT, B., BONVIN, D., 2008, "Real-time optimization via adaptation and control of the constraints". v. 25, *Computer Aided Chemical Engineering*, Elsevier, pp. 393 – 398.
- MARCHETTI, A. G., FRANÇOIS, G., FAULWASSER, T., BONVIN, D., 2016, "Modifier adaptation for real-time optimization - Methods and applications", *Processes*, v. 4, n. 4. doi: <10.3390/pr4040055>.
- MARCOS, N., GUAY, M., DOCHAIN, D., 2004, "Output feedback adaptive extremum seeking control of a continuous stirred tank bioreactor with Monod's kinetics", *Journal of Process Control*, v. 14, n. 7, pp. 807–818. doi: <10.1016/j.jprocont.2003.12.002>. Dynamics, Monitoring, Control and Optimization of Biological Systems.
- MARLIN, T., HRYMAK, A., 1997, "Real-Time Operations Optimization of Continuous Processes", *AIChE Symp Ser CPC V.*, v. 93 (01).
- MARTINEZ, E., 2000, "Batch process modeling for optimization using reinforcement learning", *Computers & Chemical Engineering*, v. 24, n. 2-7, pp. 1187–1193.
- MATIAS, J. O., LE ROUX, G. A., JÄSCHKE, J., 2018, "Modifier adaptation for real-time optimization of a gas lifted well network", *IFAC-PapersOnLine*, v. 51, n. 8, pp. 31–36. doi: <10.1016/j.ifacol.2018.06.351>.
- MATIAS, J., JÄSCHKE, J., 2019, "Using a neural network for estimating plant gradients in real-time optimization with modifier adaptation", *IFAC-PapersOnLine*, v. 52, n. 1, pp. 808 – 813. doi: <https://doi.org/10.1016/j.ifacol.2019.06.161>. Availability: <http://www.sciencedirect.com/science/article/pii/S2405896319302496>. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019.
- MATIAS, J., OLIVEIRA, J. P., LE ROUX, G. A., JÄSCHKE, J., 2022, "Steady-state real-time optimization using transient measurements on an experimental rig", *Journal of Process Control*, v. 115, pp. 181–196. doi: <https://doi.org/10.1016/j.jprocont.2022.04.015>.
- MATIAS, J. O., LE ROUX, G. A., 2018, "Real-time Optimization with persistent parameter adaptation using online parameter estimation", *Journal of Process Control*, v. 68, pp. 195 – 204. doi: <https://doi.org/10.1016/j.jprocont.2018.05.009>.

- MATIAS, J. O., LE ROUX, G. A., 2020, "Plantwide optimization via real-time optimization with persistent parameter adaptation", *Journal of Process Control*, v. 92, pp. 62–78. doi: <10.1016/j.jprocont.2020.05.006>.
- MENDOZA, D. F., GRACIANO, J. E. A., DOS SANTOS LIPORACE, F., LE ROUX, G. A. C., 2016, "Assessing the reliability of different real-time optimization methodologies", *Canadian Journal of Chemical Engineering*, v. 94, n. 3, pp. 485–497. doi: <10.1002/cjce.22402>.
- MÖNNIGMANN, M., MARQUARDT, W., 2003, "Steady-State Process Optimization with Guaranteed Robust Stability and Feasibility", *AIChE Journal*, v. 49, n. 12, pp. 3110–3126. doi: <10.1002/aic.690491212>.
- MORARI, M., ARKUN, Y., STEPHANOPOULOS, G., 1980, "Studies in the synthesis of control structures for chemical processes: Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures", *AIChE Journal*, v. 26, n. 2, pp. 220–232. doi: <10.1002/aic.690260205>.
- MOROSANOV, I., 1957, "Method of extremum control", *Automatic and Remote Control*, v. 18, pp. 1077–1092.
- MÜLLER, D., ILLNER, M., ESCHE, E., POGRZEBA, T., SCHMIDT, M., SCHOMÄCKER, R., BIEGLER, L. T., WOZNY, G., REPKE, J.-U., 2017, "Dynamic real-time optimization under uncertainty of a hydroformylation mini-plant", *Computers & Chemical Engineering*, v. 106, pp. 836–848.
- MUSKE, K. R., RAWLINGS, J. B., 1993, "Model predictive control with linear models", *AIChE Journal*, v. 39, n. 2, pp. 262–287. doi: <10.1002/aic.690390208>.
- NAJIM, K., IKONEN, E., 2001, *Advanced Process Identification and Control*. Manchester, CRC Press. ISBN: 9780367396886.
- NAVIA, D., MARTÍ, R., SARABIA, D., GUTIÉRREZ, G., DE PRADA, C., 2012, "Handling Infeasibilities in Dual Modifier-Adaptation Methodology for Real-Time Optimization", *IFAC Proceedings Volumes*, v. 45, n. 15, pp. 537 – 542. doi: <https://doi.org/10.3182/20120710-4-SG-2026.00078>. Availability: <http://www.sciencedirect.com/science/article/pii/S1474667016304980>. 8th IFAC Symposium on Advanced Control of Chemical Processes.

- NAVIA, D., BRICEÑO, L., GUTIÉRREZ, G., DE PRADA, C., 2015, "Modifier-Adaptation Methodology for Real-Time Optimization Reformulated as a Nested Optimization Problem", *Industrial & Engineering Chemistry Research*, v. 54, n. 48, pp. 12054–12071. doi: <10.1021/acs.iecr.5b01946>.
- NAYSMITH, M. R., DOUGLAS, P. L., 1995, "Review of Real Time Optimization in the Chemical Process Industries", *Developments in Chemical Engineering and Mineral Processing*, v. 3, n. 2, pp. 67–87. doi: <10.1002/apj.5500030202>.
- NIAN, R., LIU, J., HUANG, B., 2020, "A review On reinforcement learning: Introduction and applications in industrial process control", *Computers & Chemical Engineering*, v. 139, pp. 106886.
- OCHOA, S., REPKE, J.-U., WOZNY, G., 2010, "Integrating real-time optimization and control for optimal operation: Application to the bio-ethanol process", *Biochemical Engineering Journal*, v. 53, n. 1, pp. 18–25.
- PALMER, K., REALFF, M., 2002a, "Optimization and Validation of Steady-State Flowsheet Simulation Metamodels", *Chemical Engineering Research and Design*, v. 80, n. 7, pp. 773–782. doi: <https://doi.org/10.1205/026387602320776849>.
- PALMER, K., REALFF, M., 2002b, "Metamodeling Approach to Optimization of Steady-State Flowsheet Simulations: Model Generation", *Chemical Engineering Research and Design*, v. 80, n. 7, pp. 760–772. doi: <https://doi.org/10.1205/026387602320776830>.
- PANTELIDES, C., RENFRO, J., 2013, "The online use of first-principles models in process operations: Review, current status and future needs", *Computers & Chemical Engineering*, v. 51, pp. 136–148. doi: <https://doi.org/10.1016/j.compchemeng.2012.07.008>. Availability: <https://www.sciencedirect.com/science/article/pii/S0098135412002360>. CPC VIII.
- PAPASAVVAS, A., FRANÇOIS, G., 2020, "Output modifier adaptation with filter-based constraints", *Journal of Process Control*, v. 87, pp. 37–53. doi: <https://doi.org/10.1016/j.jprocont.2020.01.002>. Availability: <http://www.sciencedirect.com/science/article/pii/S0959152419305761>.

- PAPASAVVAS, A., FERREIRA, T. D. A., MARCHETTI, A. G., BONVIN, D., 2019, "Analysis of output modifier adaptation for real-time optimization", *Computers & Chemical Engineering*, v. 121, pp. 285–293. doi: <10.1016/j.compchemeng.2018.09.028>. Availability: <http://infoscience.epfl.ch/record/266936>.
- PATWARDHAN, S. C., NARASIMHAN, S., JAGADEESAN, P., GOPALUNI, B., L. SHAH, S., 2012, "Nonlinear Bayesian state estimation: A review of recent developments", *Control Engineering Practice*, v. 20, n. 10, pp. 933–953. doi: <10.1016/j.conengprac.2012.04.003>. 4th Symposium on Advanced Control of Industrial Processes (ADCONIP).
- PESCH, H. J., 1989, "Real-time computation of feedback controls for constrained optimal control problems. part 1: Neighbouring extremals", *Optimal Control Applications and Methods*, v. 10, n. 2, pp. 129–145. doi: <10.1002/oca.4660100205>.
- PIRNAY, H., LÓPEZ-NEGRETE, R., BIEGLER, L. T., 2012, "Optimal sensitivity based on IPOPT", *Mathematical Programming Computation*, v. 4, pp. 307–331. doi: <10.1007/s12532-012-0043-2>.
- PISTIKOPOULOS, E. N., BARBOSA-POVOA, A., LEE, J. H., MISENER, R., MITOSOS, A., REKLAITIS, G. V., VENKATASUBRAMANIAN, V., YOU, F., GANI, R., 2021, "Process systems engineering—the generation next?" *Computers & Chemical Engineering*, v. 147, pp. 107252.
- POLLARD, G. P., SARGENT, R. W. H., 1970, "Off Line Computation of Optimum Controls for a Plate Distillation Column", *Automatica*, v. 6, n. 1 (jan), pp. 59–76. doi: <10.1016/0005-1098(70)90075-0>.
- PONTES, K. V., WOLF, I. J., EMBIRUCU, M., MARQUARDT, W., 2015, "Dynamic real-time optimization of industrial polymerization processes with fast dynamics", *Industrial & Engineering Chemistry Research*, v. 54, n. 47, pp. 11881–11893.
- POWELL, B. K. M., MACHALEK, D., QUAH, T., 2020, "Real-time optimization using reinforcement learning", *Computers & Chemical Engineering*, v. 143, pp. 107077. doi: <https://doi.org/10.1016/j.compchemeng.2020.107077>.
- PRIOR, D., LOPEZ, S., 1999, "Grangemouth ethylene plant installs closed-loop optimization solution", *Oil & gas journal*, v. 97, n. 42, pp. 83–86.

- QUELHAS, A. D., DE JESUS, N. J. C., PINTO, J. C., 2013, "Common vulnerabilities of RTO implementations in real chemical processes", *The Canadian Journal of Chemical Engineering*, v. 91, n. 4, pp. 652–668. doi: <10.1002/cjce.21738>. Availability: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/cjce.21738>>.
- RAMANATHAN, P., MANGLA, K. K., SATPATHY, S., 2017, "Smart controller for conical tank system using reinforcement learning algorithm", *Measurement*.
- RASMUSSEN, C. E., WILLIAMS, C. K. I., 2005, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Massachusetts, The MIT Press. ISBN: 026218253X.
- RHINEHART, R. R., 2013, "Automated steady and transient state identification in noisy processes". In: *2013 American Control Conference*, pp. 4477–4493.
- RIBEIRO, L. D., SECCHI, A. R., 2019, "A Methodology to obtain analytical models that reduce the computational complexity faced in real time implementation of NMPC controllers", *Brazilian Journal of Chemical Engineering*, v. 36 (07), pp. 1255 – 1278. doi: <10.1590/0104-6632.20190363s20180457>.
- ROBERTS, P. D., 1979, "An algorithm for steady-state system optimization and parameter estimation", *International Journal of Systems Science*, v. 10, n. 7, pp. 719–734. doi: <10.1080/00207727908941614>.
- RODGER, E. A., CHACHUAT, B., 2011, "Design Methodology of Modifier Adaptation for On-Line Optimization of Uncertain Processes", *IFAC Proceedings Volumes*, v. 44, n. 1, pp. 4113 – 4118. doi: <<https://doi.org/10.3182/20110828-6-IT-1002.01055>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S1474667016442539>>. 18th IFAC World Congress.
- RODRÍGUEZ-BLANCO, T., SARABIA, D., PITARCH, J., DE PRADA, C., 2017, "Modifier Adaptation methodology based on transient and static measurements for RTO to cope with structural uncertainty", *Computers & Chemical Engineering*, v. 106, pp. 480 – 500. doi: <<https://doi.org/10.1016/j.compchemeng.2017.07.001>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0098135417302727>>. ESCAPE-26.

- ROHMAN, F. S., SATA, S. A., AZIZ, N., 2019, "Online dynamic optimization studies of autocatalytic esterification in the semi batch reactor for handling disturbance and uncertainty", *Computers & Chemical Engineering*, v. 129, pp. 106516.
- ROTAVA, O., ZANIN, A., 2005, "Multivariable control and real-time optimization - An industrial practical view", *Hydrocarbon Processing*, v. 84 (01), pp. 61-71.
- ROWE, I. H., 1970, "A bootstrap method for the statistical estimation of model parameters†", *International Journal of Control*, v. 12, n. 5, pp. 721-738. doi: <10.1080/00207177008931886>.
- SALAU, N. P., TRIERWEILER, J. O., SECCHI, A. R., MARQUARDT, W., 2009, "A New Process Noise Covariance Matrix Tuning Algorithm for Kalman Based State Estimators", *IFAC Proceedings Volumes*, v. 42, n. 11, pp. 572-577. doi: <https://doi.org/10.3182/20090712-4-TR-2008.00092>. 7th IFAC Symposium on Advanced Control of Chemical Processes.
- SALAU, N. P., TRIERWEILER, J. O., SECCHI, A. R., 2012, "State estimators for better bioprocesses operation". In: Bogle, I. D. L., Fairweather, M. (Eds.), *22nd European Symposium on Computer Aided Process Engineering*, v. 30, *Computer Aided Chemical Engineering*, Elsevier, pp. 1267-1271.
- SALAU, N. P., TRIERWEILER, J. O., SECCHI, A. R., 2014, "Observability analysis and model formulation for nonlinear state estimation", *Applied Mathematical Modelling*, v. 38, n. 23, pp. 5407-5420. doi: <https://doi.org/10.1016/j.apm.2014.03.053>.
- SANTOS, J. E. W., TRIERWEILER, J. O., FARENZENA, M., 2021, "Model Update Based on Transient Measurements for Model Predictive Control and Hybrid Real-Time Optimization", *Industrial & Engineering Chemistry Research*, v. 60, n. 7, pp. 3056-3065. doi: <10.1021/acs.iecr.1c00212>.
- SARGENT, R., SULLIVAN, G., 1978, "The Development of an Efficient Optimal Control Package". v. 7, pp. 158-168, 01. ISBN: 3-540-08708-7.
- SCHNEIDER, R., GEORGAKIS, C., 2013, "How To NOT Make the Extended Kalman Filter Fail", *Industrial & Engineering Chemistry Research*, v. 52, n. 9, pp. 3354-3362. doi: <10.1021/ie300415d>.

- SCHWENKER, F., KESTLER, H. A., PALM, G., 2001, "Three learning phases for radial-basis-function networks", *Neural Networks*, v. 14, n. 4, pp. 439 – 458. doi: <[https://doi.org/10.1016/S0893-6080\(01\)00027-2](https://doi.org/10.1016/S0893-6080(01)00027-2)>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0893608001000272>>.
- SEQUEIRA, S. E., HERRERA, M., GRAELLS, M., PUIGJANER, L., 2004, "On-line process optimization: parameter tuning for the real time evolution (RTE) approach", *Computers & chemical engineering*, v. 28, n. 5, pp. 661–672. doi: <10.1016/j.compchemeng.2004.02.024>.
- SEQUEIRA, S. E., GRAELLS, M., PUIGJANER, L., 2002, "Real-Time Evolution for On-line Optimization of Continuous Processes", *Industrial & Engineering Chemistry Research*, v. 41, n. 7, pp. 1815–1825. doi: <10.1021/ie0104641>.
- SHAH, H., GOPAL, M., 2016, "Model-Free Predictive Control of Nonlinear Processes Based on Reinforcement Learning", *IFAC-PapersOnLine*, v. 49, n. 1, pp. 89–94.
- SHAHRIARI, B., SWERSKY, K., WANG, Z., ADAMS, R. P., DE FREITAS, N., 2016, "Taking the Human Out of the Loop: A Review of Bayesian Optimization", *Proceedings of the IEEE*, v. 104, n. 1, pp. 148–175. doi: <10.1109/JPROC.2015.2494218>.
- SHAMAKI, P. B., ODLOAK, D., 2020, "Hybrid RTO with zone control MPC applied to a Gas-lift system". In: *2020 28th Mediterranean Conference on Control and Automation (MED)*, pp. 691–696. IEEE.
- SHIN, J., BADGWELL, T. A., LIU, K.-H., LEE, J. H., 2019a, "Reinforcement Learning – Overview of recent progress and implications for process control", *Computers & Chemical Engineering*, v. 127, pp. 282–294. Availability: <<https://www.sciencedirect.com/science/article/pii/S0098135419300754>>.
- SHIN, J., BADGWELL, T. A., LIU, K.-H., LEE, J. H., 2019b, "Reinforcement Learning – Overview of recent progress and implications for process control", *Computers & Chemical Engineering*, v. 127, pp. 282–294. doi: <10.1016/j.compchemeng.2019.05.029>.
- SILVER, D., HUANG, A., MADDISON, C., GUEZ, A., SIFRE, L., DRIESSCHE, G., SCHRITTWIESER, J., ANTONOGLIOU, I., PANNEERSHELVAM, V., LANCTOT, M., DIELEMAN, S., GREWE, D., NHAM, J.,

- KALCHBRENNER, N., SUTSKEVER, I., LILICRAP, T., LEACH, M., KAVUKCUOGLU, K., GRAEPEL, T., HASSABIS, D., 2016, "Mastering the game of Go with deep neural networks and tree search", *Nature*, v. 529 (01), pp. 484–489.
- SILVER, D., HUBERT, T., SCHRITTWIESER, J., ANTONOGLU, I., LAI, M., GUEZ, A., LANCTOT, M., SIFRE, L., KUMARAN, D., GRAEPEL, T., LILICRAP, T., SIMONYAN, K., HASSABIS, D., 2018, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play", *Science*, v. 362, n. 6419, pp. 1140–1144.
- SIMON, D., 2006a, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, Wiley. ISBN: 9780470045336.
- SIMON, D., 2006b, *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. ISBN: 0471708585.
- SINGHAL, M., MARCHETTI, A., FAULWASSER, T., BONVIN, D., 2018, "Active directional modifier adaptation for real-time optimization", *Computers & Chemical Engineering*, v. 115, pp. 246 – 261. doi: <<https://doi.org/10.1016/j.compchemeng.2018.02.016>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0098135418300838>>.
- SINGHAL, M., MARCHETTI, A., FAULWASSER, T., BONVIN, D., 2020, "A note on efficient computation of privileged directions in modifier adaptation", *Computers & Chemical Engineering*, v. 132, pp. 106524. doi: <<https://doi.org/10.1016/j.compchemeng.2019.106524>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0098135419304843>>.
- SINGHAL, M., MARCHETTI, A. G., FAULWASSER, T., BONVIN, D., 2017, "Improved Directional Derivatives for Modifier-Adaptation Schemes", *IFAC-PapersOnLine*, v. 50, n. 1, pp. 5718 – 5723. doi: <<https://doi.org/10.1016/j.ifacol.2017.08.1124>>. Availability: <<http://www.sciencedirect.com/science/article/pii/S2405896317316178>>. 20th IFAC World Congress.
- SKOGESTAD, S., 2000, "Self-optimizing control: The missing link between steady-state optimization and control", *Computers and Chemical Engineering*, v. 24, n. 2-7, pp. 569–575. doi: <10.1016/s0098-1354(00)00405-1>.

- SKOGESTAD, S., HALVORSEN, I. J., MORUD, J. C., 1998, "Self-optimizing control: The basic idea and taylor series analysis", .
- SORENSEN, R. C., CUTLER, C. R., 1998, "LP integrates economics into dynamic matrix control", *Hydrocarbon Processing*, v. 77 (9), pp. 57–65.
- SPEAKMAN, J., FRANÇOIS, G., 2020, "Real-time optimization via modifier adaptation of closed-loop processes using transient measurements", *Computers & Chemical Engineering*, v. 140, pp. 106969.
- SRINIVASAN, B., BONVIN, D., 2007, "Real-time optimization of batch processes by tracking the necessary conditions of optimality", *Industrial and Engineering Chemistry Research*, v. 46, n. 2, pp. 492–504. doi: <10.1021/ie0600487>.
- SRINIVASAN, B., BONVIN, D., VISSER, E., PALANKI, S., 2003, "Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty", *Computers & Chemical Engineering*, v. 27, n. 1, pp. 27 – 44.
- SRINIVASAN, B., BIEGLER, L. T., BONVIN, D., 2008, "Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function", *Computers and Chemical Engineering*, v. 32, n. 3, pp. 572–579. doi: <10.1016/j.compchemeng.2007.04.004>.
- ST-PIERRE, M., GINGRAS, D., 2004, "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system". pp. 831 – 835, 07. ISBN: 0-7803-8310-9.
- STELZER, I. V., KAGER, J., HERWIG, C., 2017, "Comparison of Particle Filter and Extended Kalman Filter Algorithms for Monitoring of Bioprocesses". In: Espuña, A., Graells, M., Puigjaner, L. (Eds.), *27th European Symposium on Computer Aided Process Engineering*, v. 40, *Computer Aided Chemical Engineering*, Elsevier, pp. 1483–1488.
- SUTTON, R. S., BARTO, A. G., 2018a, *Reinforcement Learning: An Introduction*, v. 1. 2nd ed. Massachusetts, The MIT Press.
- SUTTON, R. S., BARTO, A. G., 2018b, *Reinforcement learning: An introduction*. Cambridge, MIT press.
- SYAFIIE, S., TADEO, F., MARTINEZ, E., 2008, "Model-free learning control of chemical processes". In: *Reinforcement Learning*, InTech, London.

- TATJEWSKI, P., BRDYŚ, M. A., DUDA, J., 2001, "Optimizing control of uncertain plants with constrained feedback controlled outputs", *International Journal of Control*, v. 74, n. 15, pp. 1510–1526. doi: <10.1080/00207170110075333>.
- TATJEWSKI, P., 2002, "Iterative optimizing set-point control - The basic principle redesigned", *IFAC Proceedings Volumes*, v. 35, n. 1, pp. 49 – 54. doi: <https://doi.org/10.3182/20020721-6-ES-1901.00994>. Availability: <http://www.sciencedirect.com/science/article/pii/S1474667015394155>. 15th IFAC World Congress.
- TOSUKHOWONG, T., LEE, J. M., LEE, J. H., LU, J., 2004, "An introduction to a dynamic plant-wide optimization strategy for an integrated plant", *Computers & chemical engineering*, v. 29, n. 1, pp. 199–208.
- TRIERWEILER, J., 1997, *A systematic approach to control structure design*. Ph.D. Thesis, Universität Dortmund, 01.
- TRIERWEILER, J. O., 2014, "Encyclopedia of Systems and Control". In: *Encyclopedia of Systems and Control*, Springer-Verlag London Limited. ISBN: 9781447151029.
- TURAN, E. M., JÄSCHKE, J., 2023, "A simple two-parameter steady-state detection algorithm: Concept and experimental validation". In: Kokosis, A. C., Georgiadis, M. C., Pistikopoulos, E. (Eds.), *33rd European Symposium on Computer Aided Process Engineering*, v. 52, *Computer Aided Chemical Engineering*, Elsevier, pp. 1765–1770. Availability: <https://www.sciencedirect.com/science/article/pii/B9780443152740502808>.
- VALLURU, J., PATWARDHAN, S. C., 2019, "An Integrated Frequent RTO and Adaptive Nonlinear MPC Scheme Based on Simultaneous Bayesian State and Parameter Estimation", *Industrial & Engineering Chemistry Research*, v. 58, n. 18, pp. 7561–7578. doi: <10.1021/acs.iecr.8b05327>.
- VALLURU, J., PUROHIT, J. L., PATWARDHAN, S. C., MAHAJANI, S. M., 2015, "Adaptive Optimizing Control of an Ideal Reactive Distillation Column", *IFAC-PapersOnLine*, v. 48, n. 8, pp. 489–494. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

- VENKATASUBRAMANIAN, V., 2018, "The Promise of Artificial Intelligence in Chemical Engineering: Is it here, finally?" *AIChE Journal*, v. 65 (12).
- WÄCHTER, A., BIEGLER, L. T., 2006, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming", *Mathematical Programming*, v. 106, n. 1, pp. 25–57. doi: <10.1007/s10107-004-0559-y>.
- WENZEL, S., GAO, W., ENGELL, S., 2015, "Handling Disturbances in Modifier Adaptation with Quadratic Approximation**The research leading to these results was funded by the ERC Advanced Investigator Grant MOBOCON under the grant agreement No. 291458." *IFAC-PapersOnLine*, v. 48, n. 25, pp. 132 – 137. doi: <https://doi.org/10.1016/j.ifacol.2015.11.072>. Availability: <http://www.sciencedirect.com/science/article/pii/S2405896315023307>. 16th IFAC Workshop on Control Applications of Optimization CAO'2015.
- WENZEL, S., YFANTIS, V., GAO, W., 2017, "Comparison of regression data selection strategies for quadratic approximation in RTO". In: Espuña, A., Graells, M., Puigjaner, L. (Eds.), *27th European Symposium on Computer Aided Process Engineering*, v. 40, *Computer Aided Chemical Engineering*, Elsevier, pp. 1711 – 1716. Availability: <http://www.sciencedirect.com/science/article/pii/B9780444639653502877>.
- WILLIAMS, T. J., OTTO, R. E., 1960, "A generalized chemical processing model for the investigation of computer control", *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, v. 79, n. 5, pp. 458–473.
- WILSON, M. M. J., 2012, "APPLICATION OF REINFORCEMENT LEARNING TO BATCH DISTILLATION", .
- YELCHURU, R., SKOGESTAD, S., 2012, "Convex formulations for optimal selection of controlled variables and measurements using Mixed Integer Quadratic Programming", *Journal of Process Control*, v. 22, n. 6, pp. 995–1007. doi: <10.1016/j.jprocont.2012.04.013>.
- YING, C.-M., JOSEPH, B., 1999, "Performance and stability analysis of LP-MPC and QP-MPC cascade control systems", *AIChE Journal*, v. 45, n. 7, pp. 1521–1534. doi: <10.1002/aic.690450714>.

- YU-GENG, X., DE-WEI, L., SHU, L., 2013, "Model predictive control—status and challenges", *Acta Automatica Sinica*, v. 39, n. 3, pp. 222–236.
- ZHANG, H., ROBERTS, P., 1990, "On-line steady-state optimisation of nonlinear constrained processes with slow dynamics", *Transactions of the Institute of Measurement and Control*, v. 12, n. 5, pp. 251–261. doi: <10.1177/014233129001200504>.
- ZHANG, H., ROBERTS, P., 1991, "Integrated system optimization and parameter estimation using a general form of steady-state model", *International Journal of Systems Science*, v. 22, n. 10, pp. 1679–1693. doi: <10.1080/00207729108910744>.
- ZHANG, W.-H., FLEURY, C., 1997, "A modification of convex approximation methods for structural optimization", *Computers & Structures*, v. 64, n. 1, pp. 89 – 95. doi: <[https://doi.org/10.1016/S0045-7949\(96\)00147-2](https://doi.org/10.1016/S0045-7949(96)00147-2)>. Availability: <<http://www.sciencedirect.com/science/article/pii/S0045794996001472>>. Computational Structures Technology.
- ZHANG, Y., FORBES, J. F., 2008, "Performance Analysis of Perturbation-Based Methods for Real-Time Optimization", *The Canadian Journal of Chemical Engineering*, v. 84, n. 2, pp. 209–218. doi: <10.1002/cjce.5450840208>.
- ZHANG, Y., FRASER FORBES, J., 2000, "Extended design cost: a performance criterion for real-time optimization systems", *Comp. & Chem. Eng.*, v. 24, n. 8, pp. 1829–1841.

Appendix A

Formulation of the terminal weight of the Self-Optimizing IHMPC

The terminal cost formulation for the Infinite Horizon MPC that explicitly handles the self-optimizing variables is presented based on the early works of MUSKE and RAWLINGS (1993) and GONZÁLEZ and ODLOAK (2009).

First, we consider only the infinite quadratic portion of the original objective function of the self-optimizing IHMPC.

$$V_k = \sum_{j=1}^{\infty} \mathbf{c}_{k+j}^T \mathbf{W}_c \mathbf{c}_{k+j} \quad (\text{A.1})$$

The infinite sum of the self-optimizing variables can then be separated into two by considering the portion until $m - 1$ and the infinite portion.

$$V_k = \sum_{j=1}^{m-1} \mathbf{c}_{k+j}^T \mathbf{W}_c \mathbf{c}_{k+j} + \sum_{j=0}^{\infty} \mathbf{c}_{k+m+j}^T \mathbf{W}_c \mathbf{c}_{k+m+j} \quad (\text{A.2})$$

Applying the self-optimizing transformation, $\mathbf{c}_{k+j} = \mathbf{H} \mathbf{y}_{k+j}^h$, it is possible to return to the original output variables; and, introducing a new transformed weight matrix $\mathbf{W}_y = \mathbf{H}^T \mathbf{W}_c \mathbf{H}$, the self-optimizing infinite objective function becomes:

$$V_k = \sum_{j=1}^{m-1} (\mathbf{y}_{k+j}^h)^T \mathbf{W}_y \mathbf{y}_{k+j}^h + \sum_{j=0}^{\infty} (\mathbf{y}_{k+m+j}^h)^T \mathbf{W}_y \mathbf{y}_{k+m+j}^h \quad (\text{A.3})$$

In order to handle the infinite sum, a relation between \mathbf{y}_{k+m+j}^h and the terminal output variables \mathbf{y}_{k+m}^h is derived from the predictive model presented in Equation

4.10, which is:

$$\mathbf{y}_{k+m+j}^h = (\mathbf{A}^h)^j \mathbf{y}_{k+m}^h + \sum_{i=0}^{j-1} (\mathbf{A}^h)^i \mathbf{B}^h \mathbf{G}^d \mathbf{d}_k \quad (\text{A.4})$$

in which, the second term of the above relation will be denoted as Σ , in order to simplify notation. Substituting it in V_k , the infinite term of objective function becomes:

$$\sum_{j=0}^{\infty} [(\mathbf{A}^h)^j \mathbf{y}_{k+m}^h]^T \mathbf{W}_y (\mathbf{A}^h)^j \mathbf{y}_{k+m}^h + \sum_{j=0}^{\infty} \Sigma^T \mathbf{W}_y \Sigma \quad (\text{A.5})$$

The second sum of Equation A.5 does not depend on $\Delta \mathbf{u}$, so it has only the role of shifting the optimal value of the objective function. Since it does not affect the optimal input trajectory, this term can be neglected and removed from the analysis. With that said, the infinite sum of the objective function can be replaced by a terminal weight, becoming:

$$V_k = \sum_{j=1}^{m-1} (\mathbf{y}_{k+j}^h)^T \mathbf{W}_y \mathbf{y}_{k+j}^h + (\mathbf{y}_{k+m}^h)^T \mathbf{P} \mathbf{y}_{k+m}^h \quad (\text{A.6})$$

in which,

$$\mathbf{P} = \sum_{j=0}^{\infty} [(\mathbf{A}^h)^j]^T \mathbf{W}_y (\mathbf{A}^h)^j \quad (\text{A.7})$$

It is well known that the recurrence found for the terminal weight results in a Lyapunov equation by opening the sum in Equation A.7 and calculating $(\mathbf{A}^h)^T \mathbf{P} \mathbf{A}^h - \mathbf{P}$, which results in:

$$(\mathbf{A}^h)^T \mathbf{P} \mathbf{A}^h - \mathbf{P} - \mathbf{W}_y - [(\mathbf{A}^h)^{\infty+1}]^T \mathbf{W}_y (\mathbf{A}^h)^{\infty+1} = \mathbf{0} \quad (\text{A.8})$$

in which, the last term is equal to $\mathbf{0}$ if matrix \mathbf{A}^h is stable. Since, in the present work, \mathbf{A}^h comes from an identification problem where a stability constraint is added, see Equation 4.4, it is always stable. Returning to the original self-optimizing weight matrix, the Lyapunov equation assumes the form presented in Equation 4.20:

$$(\mathbf{A}^h)^T \mathbf{P} \mathbf{A}^h - \mathbf{P} - \mathbf{H}^T \mathbf{W}_c \mathbf{H} = \mathbf{0} \quad (\text{A.9})$$

Appendix B

Implementation of the Self-Optimizing IHMPC as a QP

Frequently, linear MPCs are formulated as QP in order to take advantage of the highly efficient solvers available to solve this kind of problem. The basic idea is to rearrange the problem in order to fit into the following problem:

$$\min_{\mathbf{x}} \mathbf{x}^T \mathbf{Q}_{qp} \mathbf{x} + \mathbf{f}_{qp}^T \mathbf{x} \quad (\text{B.1a})$$

$$\text{s.t. } \mathbf{A}_{qp} \mathbf{x} \leq \mathbf{b}_{qp} \quad (\text{B.1b})$$

in which, \mathbf{x} are the degrees of freedom of the problem, defined as $\mathbf{x} := [(\Delta \mathbf{u})^T, (\boldsymbol{\rho}_y^L)^T, (\boldsymbol{\rho}_y^U)^T, (\boldsymbol{\rho}_u^L)^T, (\boldsymbol{\rho}_u^U)^T]^T$. To start defining the QP components \mathbf{Q}_{qp} , \mathbf{f}_{qp} , \mathbf{A}_{qp} and \mathbf{b}_{qp} , first we expand the predictive model presented in Equation 4.10 by extend the control variables until a predictive horizon p and the input variables until a control horizon m :

$$\begin{bmatrix} \mathbf{y}_{k+1}^h \\ \mathbf{y}_{k+2}^h \\ \vdots \\ \mathbf{y}_{k+p}^h \end{bmatrix} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^p \end{bmatrix} \mathbf{y}_k + \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{p-1} \mathbf{B} & \mathbf{CA}^{p-2} \mathbf{B} & \dots & \mathbf{CA}^{p-m} \mathbf{B} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_k \\ \Delta \mathbf{u}_{k+1} \\ \vdots \\ \Delta \mathbf{u}_{k+m-1} \end{bmatrix} + \begin{bmatrix} \mathbf{CD} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CD} & \mathbf{CAD} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CD} & \mathbf{CAD} & \dots & \mathbf{CA}^{p-1} \mathbf{D} \end{bmatrix} \mathbf{d}_k \quad (\text{B.2})$$

Note that, the predictive and control horizons are the same for the infinite horizon MPC, that is $p = m$. Introducing new notation, Equation B.2 becomes:

$$\mathbf{y}^h = \Phi \mathbf{y}_k + \Theta \Delta \mathbf{u} + \Psi \mathbf{d}_k \quad (\text{B.3})$$

The soft constraints are added for all the output variables, in contrast with GRACIANO *et al.* (2015), which proposed to define a subset of constrained variables. So, the constraints of the output variables written in reference to the QP's degrees of freedom are:

$$-\Theta\Delta\mathbf{u} - \boldsymbol{\rho}_y^L \leq \tilde{\mathbf{I}}^y \mathbf{y}^L - \Phi \mathbf{y}_k - \Psi \mathbf{d}_k \quad (\text{B.4a})$$

$$\Theta\Delta\mathbf{u} - \boldsymbol{\rho}_y^U \leq \Phi \mathbf{y}_k + \Psi \mathbf{d}_k - \tilde{\mathbf{I}}^y \mathbf{y}^U \quad (\text{B.4b})$$

in which,

$$\tilde{\mathbf{I}}^y = \begin{bmatrix} \mathbf{I}_{n_y} \\ \mathbf{I}_{n_y} \\ \vdots \\ \mathbf{I}_{n_y} \end{bmatrix} \in \mathbb{R}^{(pn_y) \times n_y} \quad (\text{B.5})$$

In addition, the soft constraints added to the input variables are:

$$-\tilde{\mathbf{M}}\Delta\mathbf{u} - \boldsymbol{\rho}_u^L \leq \tilde{\mathbf{I}}^u (\mathbf{u}_{k-1} - \mathbf{u}^L) \quad (\text{B.6a})$$

$$\tilde{\mathbf{M}}\Delta\mathbf{u} - \boldsymbol{\rho}_u^U \leq \tilde{\mathbf{I}}^u (\mathbf{u}^U - \mathbf{u}_{k-1}) \quad (\text{B.6b})$$

in which,

$$\tilde{\mathbf{I}}^u = \begin{bmatrix} \mathbf{I}_{n_u} \\ \mathbf{I}_{n_u} \\ \vdots \\ \mathbf{I}_{n_u} \end{bmatrix} \in \mathbb{R}^{(mn_u) \times n_u} \quad (\text{B.7})$$

and

$$\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{I}_{n_u} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I}_{n_u} & \mathbf{I}_{n_u} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{n_u} & \mathbf{I}_{n_u} & \dots & \mathbf{I}_{n_u} \end{bmatrix} \in \mathbb{R}^{(mn_u) \times (mn_u)} \quad (\text{B.8})$$

Therefore, the QP matrices \mathbf{A}_{qp} and \mathbf{b}_{qp} are:

$$\mathbf{A}_{qp} = \begin{bmatrix} -\Theta & -\mathbf{I}_{pn_y} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \Theta & \mathbf{0} & -\mathbf{I}_{pn_y} & \mathbf{0} & \mathbf{0} \\ -\tilde{\mathbf{M}} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_{mn_u} & \mathbf{0} \\ \tilde{\mathbf{M}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_{mn_u} \end{bmatrix} \quad (\text{B.9})$$

$$\mathbf{b}_{qp} = \begin{bmatrix} \tilde{\mathbf{I}}^y \mathbf{y}^L - \Phi \mathbf{y}_k - \Psi \mathbf{d}_k \\ \Phi \mathbf{y}_k + \Psi \mathbf{d}_k - \tilde{\mathbf{I}}^y \mathbf{y}^U \\ \tilde{\mathbf{I}}^u (\mathbf{u}_{k-1} - \mathbf{u}^L) \\ \tilde{\mathbf{I}}^u (\mathbf{u}^U - \mathbf{u}_{k-1}) \end{bmatrix} \quad (\text{B.10})$$

The quadratic objective function is implemented as a multi-objective compromise between tracking the setpoints of the self-optimizing variables and penalizing the movements of the input variables and the violation of the soft constraints, such that:

$$J_k = (\mathbf{y} - \mathbf{y}^{sp})^T \tilde{\mathbf{W}}_y (\mathbf{y} - \mathbf{y}^{sp}) + \Delta \mathbf{u}^T \tilde{\mathbf{W}}_{\Delta u} \Delta \mathbf{u} + \mathbf{w}_\rho^T \boldsymbol{\rho} \quad (\text{B.11})$$

in which, $\boldsymbol{\rho} := [(\boldsymbol{\rho}_y^L)^T, (\boldsymbol{\rho}_y^U)^T, (\boldsymbol{\rho}_u^L)^T, (\boldsymbol{\rho}_u^U)^T]^T$, \mathbf{w}_ρ is the vector of weights for each slack variables, $\mathbf{y}^{sp} = [(\mathbf{y}_{k+1}^{sp})^T, \dots, (\mathbf{y}_{k+p}^{sp})^T]^T$, $\tilde{\mathbf{W}}_{\Delta u} = \text{diag}([\mathbf{W}_{\Delta u}, \dots, \mathbf{W}_{\Delta u}]) \in \mathbb{R}^{(mn_u) \times (mn_u)}$ and $\tilde{\mathbf{W}}_y = \text{diag}([\mathbf{W}_y, \dots, \mathbf{W}_y, \mathbf{W}_y + \mathbf{P}]) \in \mathbb{R}^{(mn_y) \times (mn_y)}$. Although the self-optimizing variables do not appear explicitly in Equation B.11, they are naturally handled by the transformed weight matrix $\mathbf{W}_y = \mathbf{H}^T \mathbf{W}_c \mathbf{H}$.

The QP matrices \mathbf{Q}_{qp} and \mathbf{f}_{qp} are:

$$\mathbf{Q}_{qp} = \begin{bmatrix} \Theta^T \tilde{\mathbf{W}}_y \Theta + \tilde{\mathbf{W}}_{\Delta u} \\ \mathbf{0} \end{bmatrix} \quad (\text{B.12})$$

$$\mathbf{f}_{qp} = \begin{bmatrix} ((\Phi \mathbf{y}_k + \Psi \mathbf{d}_k - \mathbf{y}^{sp})^T \tilde{\mathbf{W}}_y \Theta)^T \\ \mathbf{w}_\rho \end{bmatrix} \quad (\text{B.13})$$

Appendix C

Supplementary Material for Chapter 4: Closed-loop performance of the Adaptive SOC Framework in the presence of measurement noise

This Supplementary Material analyses the effect of measurement noise on the proposed HRTO with an adaptive Self-Optimizing Control framework. To do so, a zero-mean normal distributed noise was artificially added to the measurements in the following manner:

$$\bar{\mathbf{y}}_k^m = [\mathbf{I}_{n_y} + \text{diag}(\mathbf{w}_y)] \bar{\mathbf{y}}_k^0 \quad (\text{C.1})$$

in which, $\mathbf{w}_y \in \mathbb{R}^{n_y} \sim \mathcal{N}(\mathbf{0}, \Sigma_y)$ and Σ_y is the measurement noise covariance matrix. This covariance matrix is considered unknown, but it was set to $\Sigma_y = n\mathbf{I}_{n_y}$, considering that n is the noise amplitude.

The results presented here are very similar to the results in Section 4.3 of the main paper, with the difference that the effect of noise is observed. The same simulation scenario is compared for the schemes: HRTO using HEKF tuning 1, HRTO using HEKF tuning 2, and RTO for Hammerstein models 1 and 2. Moreover, two noise amplitudes are compared: $n = 1\%$ and $n = 5\%$. Figures C.1 and C.2 show the results of the input and output variables considering model 1 with 1% noise amplitude, respectively.

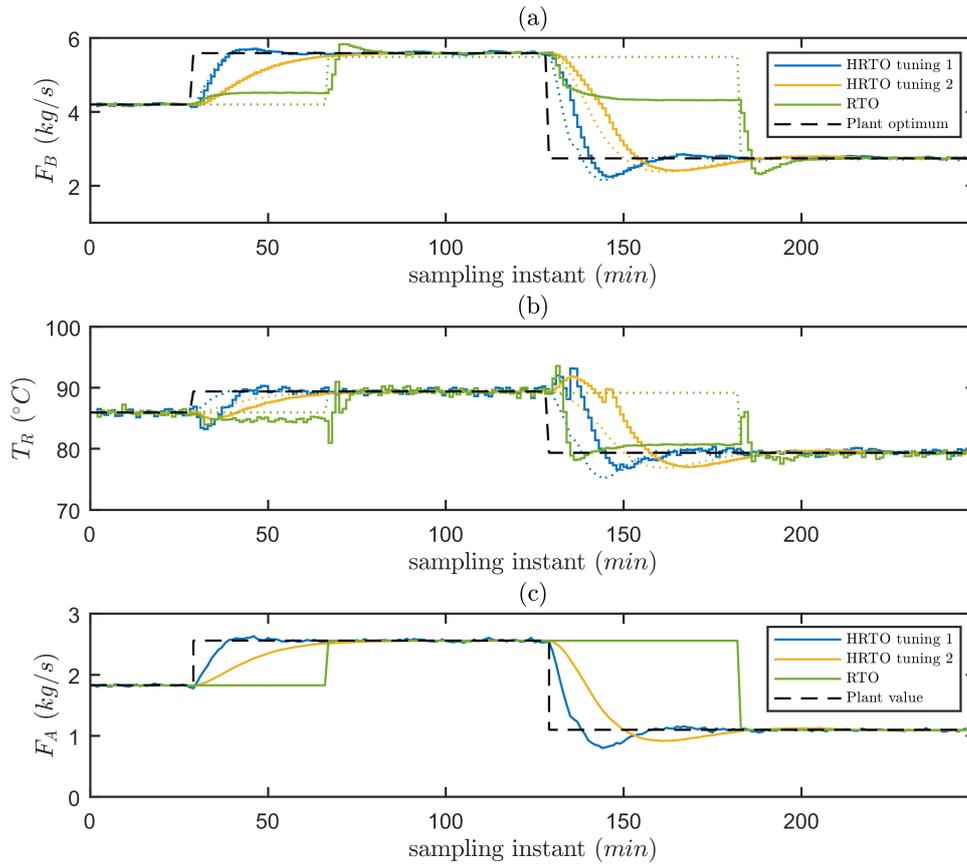


Figure C.1: Closed-loop performance using Hammerstein model 1 in the presence of 1% measurement noise: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal inputs obtained by the HRTO.

In the presence of noisy measurements, it is clear that, for HRTO tuning 1 and RTO, noise is propagated through the control layer and reflects on noisy optimal control actions, which is better visualized from the results of variable T_R . On the other hand, HRTO tuning 2 can filter noise, resulting in a smoother trajectory in the control actions. However, the better noise filtering ability of the scheme using tuning 2 comes with the cost of a slower convergence of the unmeasured variable to its true value. HRTO tuning 1 presented a faster convergence of the unmeasured parameter. However, the measurement noise is also propagated to the estimated value, resulting in the noise propagation through the model switching strategy and the nonlinear economic optimization problem solution.

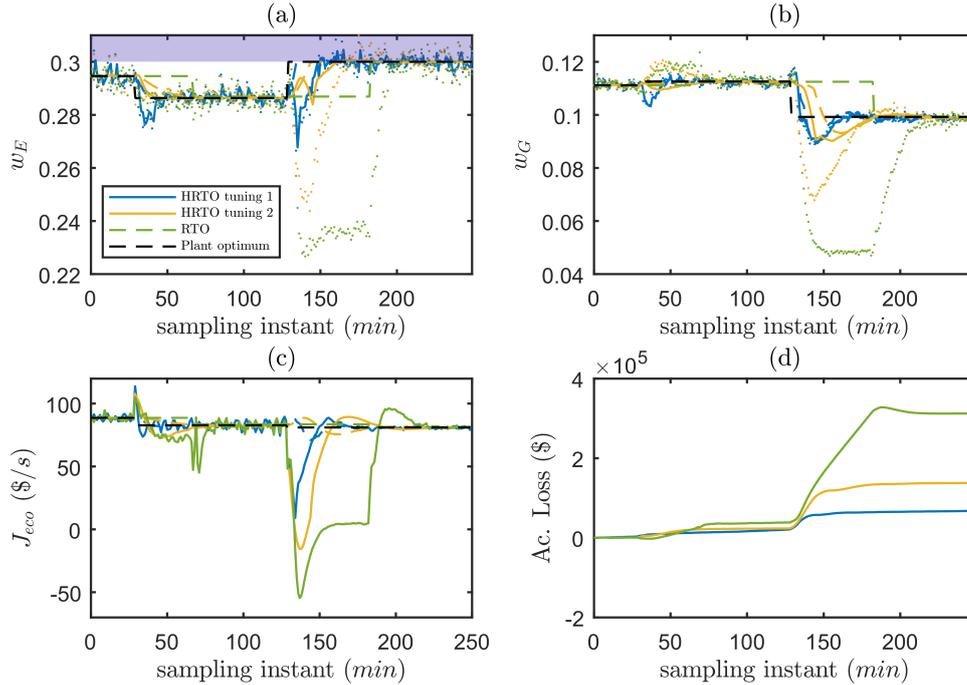


Figure C.2: Closed-loop performance using Hammerstein model 1 in the presence of 1 % measurement noise: (a) w_G ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function (\$). Dots represent the noisy measurements, solid lines represent quantities estimated by the HEKF, dashed lines represent the optimal setpoints obtained by the HRTO, and the grey area represents the violation of the imposed constraint.

The noise effect is also prominent in the output variables. HRTO tuning 2 performs better in smoothing the measurement noise, but slower parameter convergence time leads to a worse economic performance than HRTO tuning 1. That also translates to the RTO scheme, considering that long waiting times for SSD lead to long periods of suboptimal performance. Another important point to mention is the frequent violations of the active constraint due to the noise. In schemes HRTO tuning 1 and RTO, the estimated quantities are also subjected to this effect due to the inability to reject noise, which is not the case for HRTO tuning 2. These effects are better observed when a higher noise amplitude is considered, and even other effects emerge.

Figures C.3 and C.4 show the results of the input and output variables for the frameworks considering model 1 with 5 % noise amplitude, respectively.

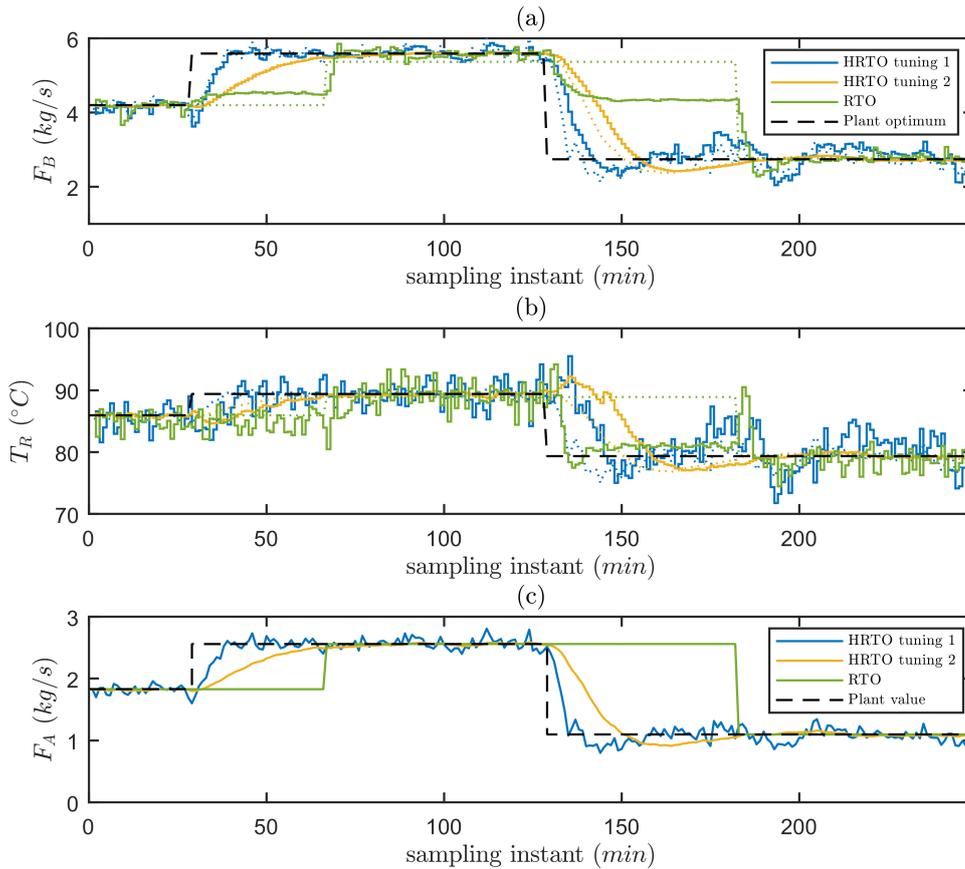


Figure C.3: Closed-loop performance using Hammerstein model 1 in the presence of 5 % measurement noise: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal inputs obtained by the HRTO.

In the case considering 5 % noise amplitude, the inputs are considerably noisier than the case with 1 %, especially for HRTO tuning 1, which also presents a high noise level in the estimated parameter. In this case, the high convergence speed is overshadowed by its noise level, compromising the economic benefit of the parameter updating proposed by HRTO methodology, as shown in Figure C.4(d). On the other hand, HRTO tuning 2 can still filter the noise even in this more challenging noise scenario, resulting in smooth parameter estimation and input trajectories.

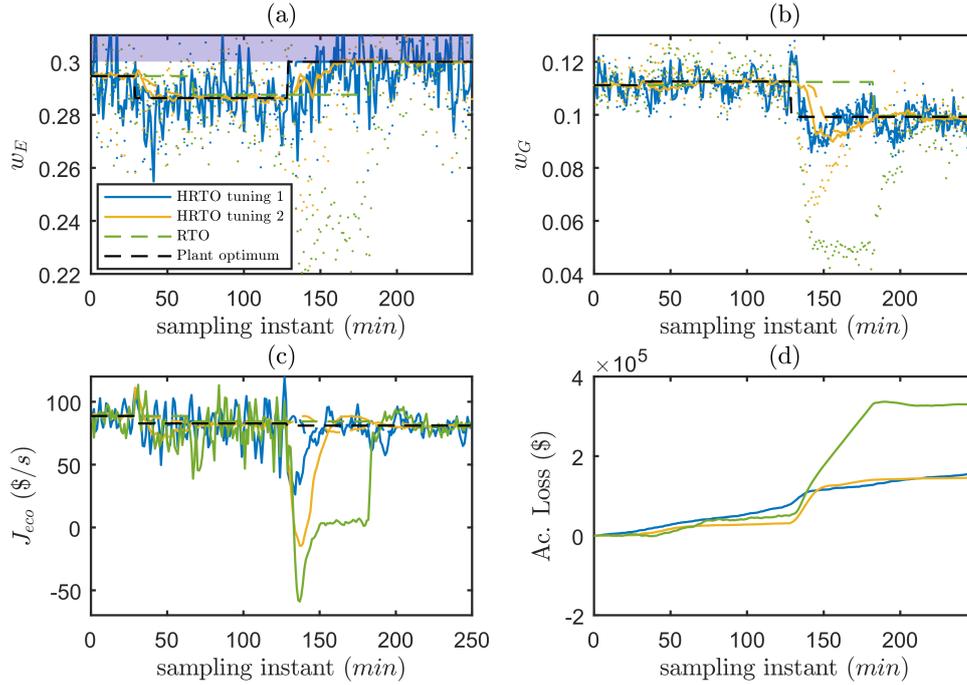


Figure C.4: Closed-loop performance using Hammerstein model 1 in the presence of 5 % measurement noise: (a) w_G ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function (\$). Dots represent the noisy measurements, solid lines represent quantities estimated by the HEKF, dashed lines represent the optimal setpoints obtained by the HRTO, and the grey area represents the violation of the imposed constraint.

It is interesting to note that Figure C.4(d) presents three distinct periods. In the first period, from 0 *min* to about 60 *min*, it is possible to see that the RTO presented the best economic performance. This suggests that not updating the parameter could be an interesting strategy for the first disturbance since the framework relies on the SOC's ability to minimize the loss function by controlling combinations of the measurement variables that present minimum sensibility to disturbances and measurement noise. During this period, HRTO tuning 1 presented the worst result due to its noise filtering inability. The RTO economic performance was even better than the HRTO tuning 2. This fact implies that the dynamic parameter convergence was economically unfavorable. However, after converging to the parameter true value, the inputs were droved to their optimal values. The HRTO tuning 2 economic performance surpassed the RTO, as observed in the following period from 60 *min* to 125 *min*. In this second period, it is possible to see that the RTO economic performance remained better than the HRTO tuning 1 and worse than HRTO tuning 2. In the third and final period, starting from about 125 *min*, the economic benefit of the HRTO becomes most evident. In this period, the subjected disturbance is responsible for activating the constraint, and the cost of the suboptimal operation in this scenario droved the RTO to the worst economic sce-

nario. In addition, the HRTO tuning 2 performance approached the HRTO tuning 1 due to the slow convergence time to the true parameter. Moreover, the HRTO tuning 1 best performance observed for the 1 % noise amplitude is not observed in this case.

Figures C.5 and C.6 present the results of the three schemes considering the use of the Hammerstein model 2 and 1 % noise.

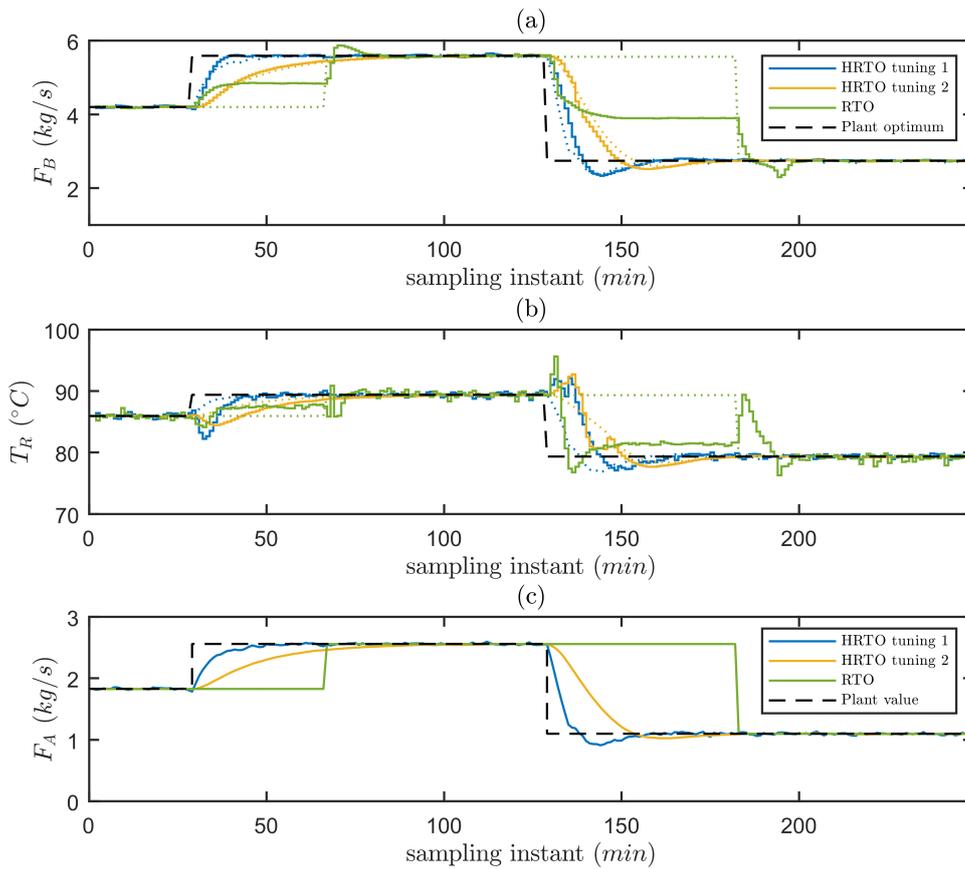


Figure C.5: Closed-loop performance using Hammerstein model 2 in the presence of 1 % measurement noise: (a) F_B (kg/s); (b) T_R ($^{\circ}C$); (c) F_A (kg/s). The dotted lines represent the optimal setpoints obtained by the HRTO.

The same outcome is observed using model 2. HRTO tuning 2 presents a higher noise filtering capacity, preventing its propagation through the parameter estimation, control, and optimization layers. This effect leads to smoother results, resulting in a smaller input variability than HRTO tuning 1 and RTO. However, this comes with the cost of a slower convergence of the unmeasured disturbance to its true value compared to HRTO tuning 1.

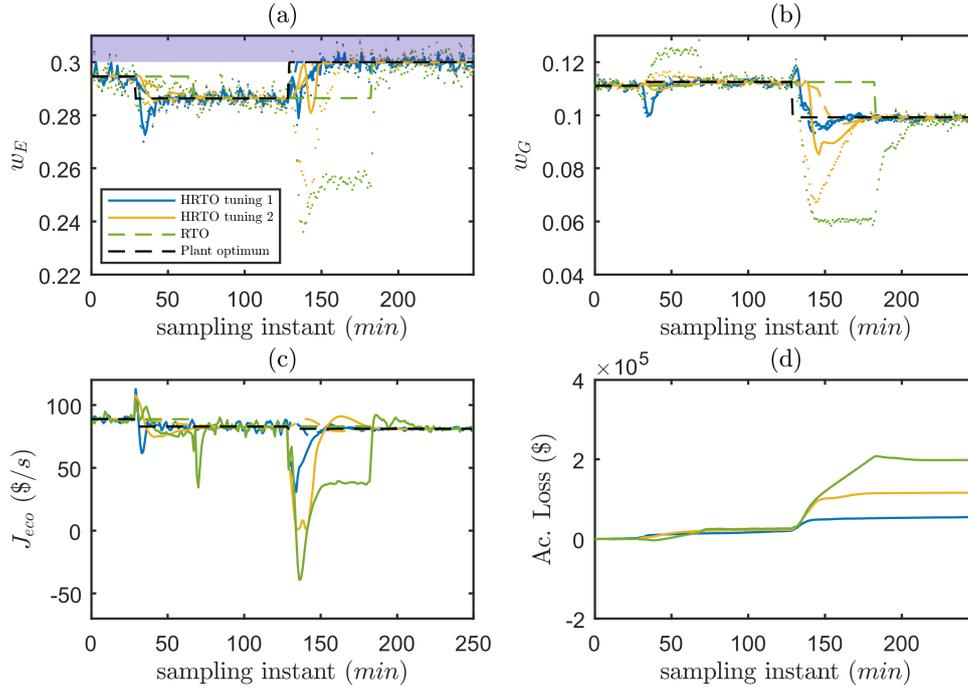


Figure C.6: Closed-loop performance using Hammerstein model 2 in the presence of 1 % measurement noise: (a) w_G ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function (\$). Dots represent the noisy measurements, solid lines represent quantities estimated by the HEKF, dashed lines represent the optimal setpoints obtained by the HRTO, and the grey area represents the violation of the imposed constraint.

The difference observed by comparing the use of model 2 over model 1 is a better description of the plant dynamics by model 2. The input variables can follow the optimal trajectory more closely, and the convergence of the unmeasured disturbance to its true value occurs more rapidly. However, HRTO tuning 1 is still faster than HRTO tuning 2. All the observations regarding the effect of noise previously made for model 1 are also applicable to model 2.

As mentioned in Section 4.3, model 2 outperforms model 1. However, one should notice that model 1 presents n_y times fewer parameters to be estimated over model 2, which is an important advantage for large-scale systems. Moreover, HRTO tuning 1 presented a better economic performance than HRTO tuning 2. However, HRTO tuning 2 presented the ability to soften the noise, which is an operational advantage, which enhances robustness and reduces the variability of the framework.

Figures C.7 and C.8 present the result of the three schemes considering the use of the Hammerstein model 2 and 5 % noise.

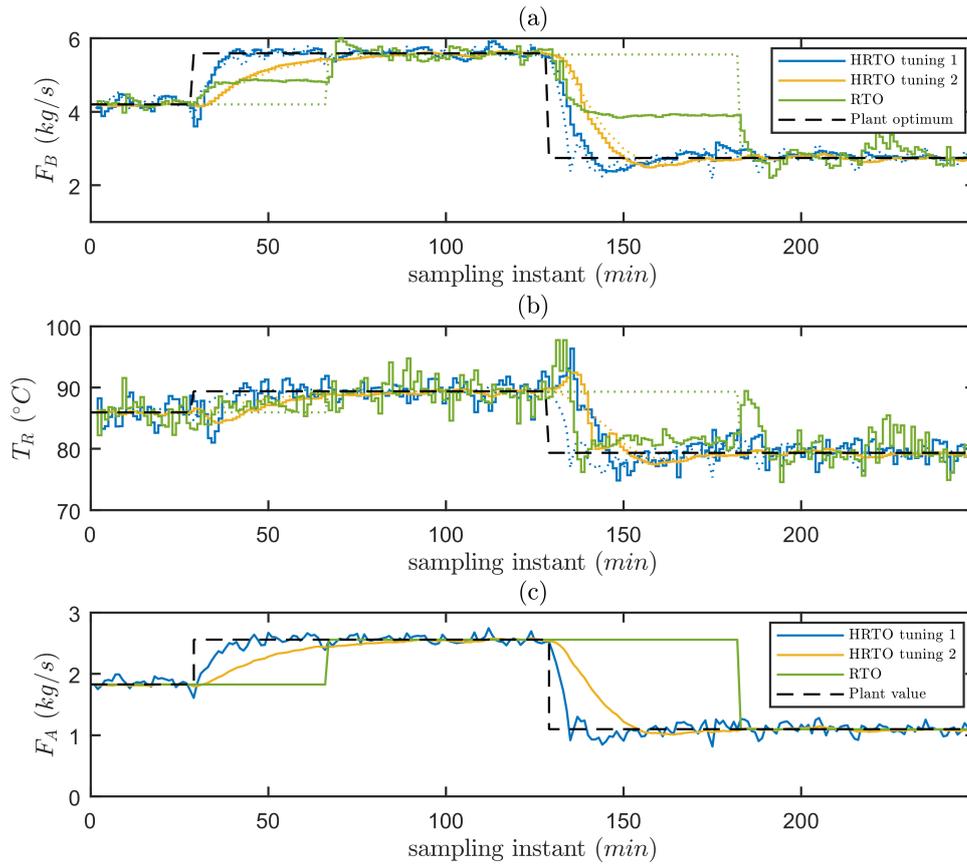


Figure C.7: Closed-loop performance using Hammerstein model 2 in the presence of 5 % measurement noise: (a) F_B (kg/s); (b) T_R (°C); (c) F_A (kg/s). The dotted lines represent the optimal inputs obtained by the HRTO.

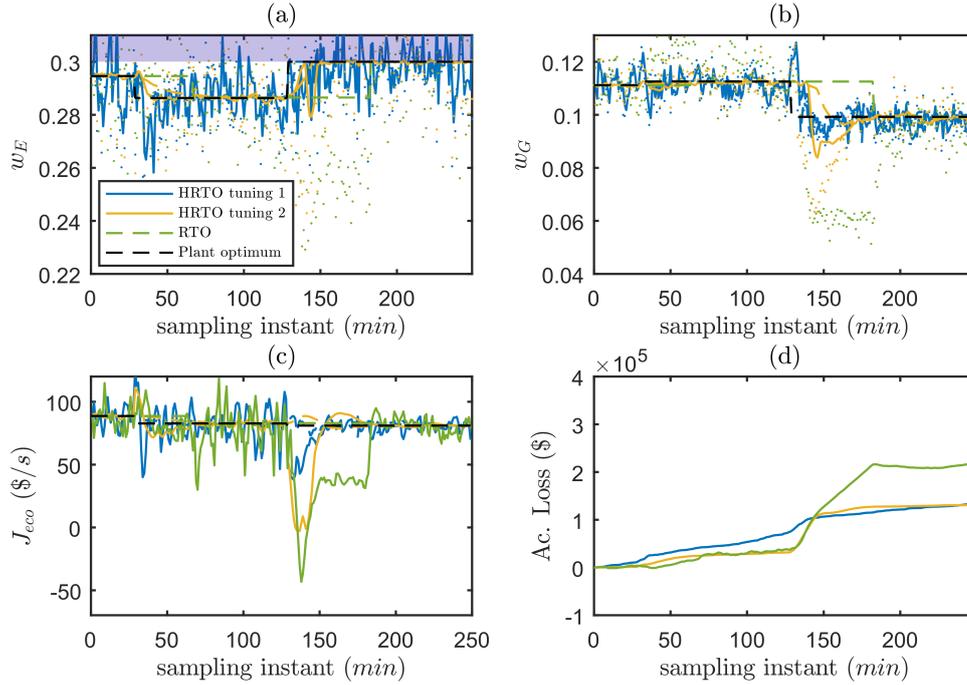


Figure C.8: Closed-loop performance using Hammerstein model 2 in the presence of 5 % measurement noise: (a) w_G ; (b) w_P ; (c) Objective function, $J_{eco}(\$/s)$; (d) Accumulated loss function (\$). Dots represent the noisy measurements, solid lines represent quantities estimated by the HEKF, dashed lines represent the optimal setpoints obtained by the HRTO, and the grey area represents the violation of the imposed constraint.

The same observation previously made for model 1 could also be done for the results using model 2. The use of model 2 enhances the accuracy of the dynamic model. Despite that, HRTO tuning 1 also presented the worst economic performance previous to the second disturbance and approached HRTO tuning 2 performance after it. However, it is important to highlight that another tuning that better balances the trade-off between noise filtering and parameter convergence speed could be used, but we kept tunings 1 and 2 for the sake of comparison.

Appendix D

The Reinforcement Learning Problem

In an RL problem, an agent learns a specific task by direct interaction with its environment (SUTTON and BARTO, 2018b).

The interaction between the agent and its environment is modeled as a Markov decision process (MDP), that SUTTON and BARTO (2018b) defined by a tuple (S, A, P, R) , where S is the state space, A is the action space, P is the stochastic state transition function $P : S \times A \times S \rightarrow [0, 1]$, and R is the reward function $R : S \times A \times S \rightarrow \mathbb{R}$. The state $s \in S$ is some information available to the agent about its environment, and the state space S is the set of all possible states. The state is assumed to be Markov, meaning that the next state is independent of the past states given the present one, as expressed by the following property (SUTTON and BARTO, 2018b):

$$\Pr[s_{t+1} | s_t] = \Pr[s_{t+1} | s_1, s_2, \dots, s_t] \quad (\text{D.1})$$

The action space A is the set of possible actions the agent may implement upon the environment. The stochastic state transition function P gives the probability for the current state $s \in S$ to transition to state $s' \in S$ when action $a \in A$ is taken by the agent:

$$P_{ss'}^a = \Pr[s_{t+1} = s' | s_t = s, a_t = a] \quad (\text{D.2})$$

The reward function R gives the immediate reward obtained by the agent by taking action a when state s transitions to state s' :

$$r = R(s, a, s') \quad (\text{D.3})$$

In the context of process control, the terms agent, environment, and action can be

interpreted as controller, process to be controlled, and control signal, respectively. When learning, at time instant t , the agent takes an action $a_t \in A$ under which the environment moves from state $s_t \in S$ to state $s_{t+1} \in S$, and receives an immediate reward $r_{t+1} \in \mathbb{R}$, to give the agent an idea of how well it is performing. The objective of the agent is to learn to take the actions that maximize the expected value of the cumulative sum of future rewards from a given state $s \in S$. The behavior of the agent at a given time is defined by a policy $\pi : S \times A \rightarrow [0, 1]$, which is a mapping from states to probabilities of selecting each possible action. In other words, if the agent is following policy π at time instant t , then $\pi(a | s)$ is the probability that $a_t = a$ if $s_t = s$. In the deterministic case, the policy becomes a mapping from states to actions: if the agent is following the policy π at time instant t , then $a_t = \pi(s)$ if $s_t = s$. The expected value of the cumulative sum of future rewards following a given policy π from a given state $s \in S$ is defined by the following state-value function:

$$V_\pi(s) = E_\pi \left\{ \sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k}, s_t = s \right\} \quad (\text{D.4})$$

where $\gamma \in [0, 1]$ is a discount factor that is a measure of how far-sighted the agent is regarding the rewards. Let us note that for the control of continuous tasks, one must have that $\gamma < 1$ to guarantee that the value function is limited.

Similarly, the expected sum of future rewards following a given policy π from a given state $s \in S$ when the action $a \in A$ is taken is defined by the following action-value function:

$$Q_\pi(s, a) = E_\pi \left\{ \sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k}, s_t = s, a_t = a \right\} \quad (\text{D.5})$$

A fundamental property of these value functions in the context of RL is that they satisfy the Bellman Expectation Equations shown in Equations D.6 and D.7. These recursive relationships form the basis of several iterative algorithms to learn V_π and Q_π from learning episodes (SUTTON and BARTO, 2018b).

$$V_\pi(s) = E_\pi[r_{t+1} + \gamma V_\pi(s_{t+1}), s_t = s] \quad (\text{D.6})$$

$$Q_\pi(s, a) = E_\pi[r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1}), s_t = s, a_t = a] \quad (\text{D.7})$$

In an RL problem, the objective may be expressed as to find the policy π^* that maximizes the state-value function with $V^*(s) = \max_{\pi} V_\pi(s)$ or the action-value function with $Q^*(s, a) = \max_{\pi} Q_\pi(s, a)$. Note that, in the latter case, the optimal policy can be produced directly from $Q^*(s, a)$ by solving the optimization problem

defined in Equation D.8.

$$\pi^*(s) = \max_a Q^*(s, a) \quad (\text{D.8})$$

The Bellman Expectation Equations shown in Equations D.6 and D.7 applied to the optimal state and action-value functions are the Bellman Optimality Equations shown in Equations D.9 and D.10. The resolution of an RL problem and the learning of the optimal policy relies on finding the solution to the Bellman Optimality Equation. The following section presents the main approaches that can be employed to address this objective.

$$V^*(s) = \max_a E[r_{t+1} + \gamma V^*(s_{t+1}), s_t = s, a_t = a] \quad (\text{D.9})$$

$$Q^*(s, a) = E[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a'), s_t = s, a_t = a] \quad (\text{D.10})$$

The resolution of the Bellman Optimality Equation

Equations D.9 and D.10 cannot be solved analytically. However, the recursive relationship between the value functions allowed the development of iterative algorithms for its resolution. Dynamic Programming considers the case where the state transition function P is known, and Reinforcement Learning deals with the case where the environment dynamics are not known a priori. In RL, the agent learns through its experiences of interaction with the environment, referred to as episodes. The simplest approach is the value-based approach where the value function is learned, and the policy is directly generated from selecting the actions that maximize the action-value function. More specifically, the following iterative process is implemented until convergence of the policy:

- Policy evaluation: the value function associated with the current policy is estimated from the mean of the accumulated rewards, denominated returns, obtained in the pairs state-action visited by the agent during the learning episodes;
- Policy improvement: the current policy is improved by selecting the actions that maximize the current value function estimate by solving the optimization problem defined in Equation D.11.

$$\pi(s) = \max_a Q(s, a) \quad (\text{D.11})$$

For low-dimension finite MDPs, solving the optimization problem defined in Equation D.11 is straightforward as the solution is the maximum value among

a finite set of real numbers. However, with continuous state and action spaces, this problem becomes a complex Nonlinear Programming (NLP) problem. Discretization may be considered, but the RL problem is then confronted with the curse of dimensionality: the data needed for training increases exponentially with the number of pairs (state, action), and the discretization rate.

To address continuous problems without discretization and the resolution of the NLP optimization problem defined in Equation D.11, policy-based methods may be applied. In this approach, the policy is explicitly represented by a parameterized approximator, and policy-gradient algorithms are used to directly learn the policy. The policy is iteratively improved by gradient ascent towards an improvement of the value function where the gradient is estimated using the cumulative rewards obtained during the experience episodes. Because of their high variance, the improvement of the policy is slower, and consequently less efficient than value-based methods.

Instead of using the cumulative rewards, estimates of the value function may then be used to reduce variance, to give rise to the actor-critic methods, where both policy and value function are learned. The actor is the component that learns the policy while the critic keeps evaluating the value function associated with the current policy to criticize the actor's action choices. This class of methods is nowadays the most popular approach to tackle continuous MDPs as they combine the best features of the two other classes of methods (SHIN *et al.*, 2019a). To represent the value function and the policy, linear combinations of state-dependent RBFs, neural networks, and kernel-based approximators may be used (BUSONI *et al.*, 2010).

The actor-critic algorithm

The idea behind the actor-critic algorithm is described in Figure D.1.

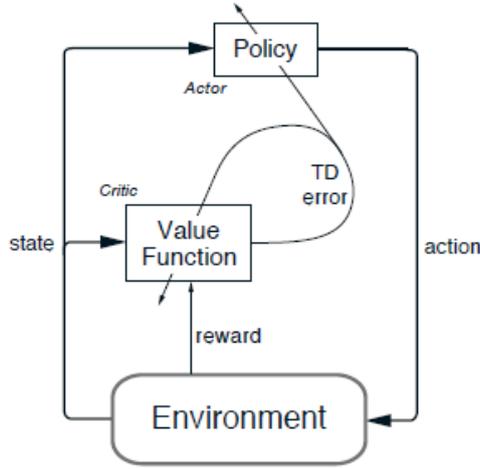


Figure D.1: The actor-critic framework (SUTTON and BARTO, 2018b). License: creativecommons.org/licenses/by-nc-nd/2.0/legalcode.

At each time step t , from the state $s_t \in S$ representing the environment, the actor selects and implements an action $a_t \in A$ upon which the environment evolves to a new state $s_{t+1} \in S$. From this transition, the immediate reward $r_{t+1} \in \mathbb{R}$ is obtained, and a new estimate of $V(s_t)$ is then $r_{t+1} + \gamma V(s_{t+1})$. The critic evaluates how well the agent went using the Temporal Difference (TD) error defined in Equation D.12, which is the difference between the new and current estimates of $V(s_t)$. If this value is positive, it indicates that the choice of a_t should be strengthened in the future, and weakened if negative (SUTTON and BARTO, 2018b).

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \quad (\text{D.12})$$

Then, the TD error is used by the critic to update the value function parameters towards its minimization, and by the actor to update the policy parameters towards the maximization of the value function. This procedure is repeated until the convergence of the policy.

Policy evaluation methods

Consider the learning episode represented by Equation D.13.

$$s_1, a_1, s_2, r_2, a_2, s_3, r_3, a_3, \dots, s_{T-1}, r_{T-1}, a_{T-1}, s_T, r_T \quad (\text{D.13})$$

where $s_T \in S$ is the episode terminal state. From this sequence, the state-value function associated with the states visited in the episode is updated using new estimates. Different options may be considered for their computation depending on how many steps ahead the rewards from the episode are considered. In that regard, the n -step ahead return from state $s_t \in S$, defined in Equation D.14, is

the sum of the discounted sum of the n consecutive rewards of the episode and the value function associated with state s_{t+n} . The higher the value of n , the more episode rewards will be taken into account, and consequently, the more the current episode will impact the value function update (SUTTON and BARTO, 2018b).

$$G_t^{(n)} = \sum_{k=1}^n \gamma^{k-1} r_{t+k} + \gamma^n V(s_{t+n}) \quad (\text{D.14})$$

The n -step ahead return defined in Equation D.14 is then used to update the value function associated with state s_t using Equation D.15.

$$V(s_t) \leftarrow V(s_t) + \alpha(G_t^{(n)} - V(s_t)) \quad (\text{D.15})$$

where $\alpha \in \mathbb{R}$ is the learning rate.

Particular cases are:

- The Monte Carlo (MC) evaluation method for which $n = T - t - 1$, and all the subsequent rewards of the episodes are considered. For this reason, it has the advantage of being unbiased. However, it is not applicable for nonending episodes (e.g. the control of continuous processes) as it requires the episode to end for the value function update. Moreover, the high variance of the MC returns leads to slow convergence properties (NIAN *et al.*, 2020; SUTTON and BARTO, 2018b).
- The TD evaluation method is the case where $n = 1$. For only considering the one-step ahead return, this evaluation method is biased, has a low variance, converges faster than the MC method, and has the great advantage of allowing the online update of the value function since it can be implemented at each time step of the episode (NIAN *et al.*, 2020).

Between these two extremes, there is a spectrum of possibilities that could be beneficial for learning. To benefit from each of them, the forward view TD(λ) policy evaluation method considers the λ -return, which is a weighted mean of all the n -step ahead returns for the estimation of the value function $V(s_t)$, as shown in Equation D.16 (SUTTON and BARTO, 2018b).

$$G_t^\lambda = (1 - \lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_t^{(n)} + \lambda^{T-t-1} G_t^{(T-t)} \quad (\text{D.16})$$

where $\lambda \in [0, 1]$ is a tuning parameter.

However, the use of G_t^λ for the value function evaluation still has the disadvantage of requiring the end of the episode to be computed. To address this issue,

an equivalent backward view TD(λ) policy evaluation method was developed to allow an online update of the value function. With this methodology, the state-value function associated with each $s \in S$ is updated towards the current TD error at a rate dependent on an eligibility trace function of the corresponding state $E(s)$. It is defined in a way to attribute more responsibility to the most recent and visited states on the value of the last obtained reward.

At the beginning of each episode, the eligibility trace of each state $s \in S$ is initialized to 0, $E_0(s) = 0$. Then, at each time step t of the episode, the TD error is computed using Equation D.12, and the eligibility trace of each $s \in S$ is updated according to Equation D.17 (SUTTON and BARTO, 2018b).

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(s = s_t) \quad (\text{D.17})$$

where λ is the eligibility trace decay parameter. The first term of the right-hand side of equation D.17 decays at $\gamma\lambda$ rate, and allows the agent to forget states that haven't been visited for some time. The second term adds one unit to the eligibility trace of the current state and allows to give responsibility to the most recently visited states in the update of the value function. Then, for each state $s \in S$, the state-value function is updated towards the TD error at a $\alpha E(s)$ rate according to Equation D.18 (SUTTON and BARTO, 2018b).

$$V(s) \leftarrow V(s) + \alpha \delta_t E(s) \quad (\text{D.18})$$

Exploration

Independently of the algorithm, the optimal policy will be obtained provided that the spaces of states and actions are sufficiently explored by the agent, otherwise, sub-optimal policies may be obtained. As a consequence, it is important to implement an exploration mechanism to guarantee that all actions and states are sufficiently considered during the training. The convergence to the optimal policy will be achieved if enough data is available and sufficient exploration is implemented. Taking random action with a certain probability (SUTTON and BARTO, 2018b) or adding white noise to the control actions (BRADTKE *et al.*, 1994) are common ways for the agent to implement exploration during training.

It is also important that the exploration decreases with time as training is undergone in order to allow the policy to improve by being greedy regarding the value function maximization, and finally converge to its optimal value.