



UM SISTEMA INTELIGENTE DE SEGURANÇA DE PROCESSOS QUÍMICOS

Rafael Marinho Soares

Dissertação de Mestrado 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 Mestre em Engenharia Química.

Orientadores: José Carlos Costa da Silva Pinto
Argimiro Resende Secchi

Rio de Janeiro
Junho de 2014

UM SISTEMA INTELIGENTE DE SEGURANÇA DE PROCESSOS QUÍMICOS

Rafael Marinho Soares

DISSERTAÇÃO 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 MESTRE EM CIÊNCIAS EM ENGENHARIA QUÍMICA.

Examinada por:

Prof. José Carlos Costa da Silva Pinto, D.Sc.

Prof. Argimiro Resende Secchi, D.Sc.

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

Prof. Márcia Peixoto Vega Domiciano, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

JUNHO DE 2014

Soares, Rafael Marinho

Um sistema inteligente de segurança de processos químicos/Rafael Marinho Soares. – Rio de Janeiro: UFRJ/COPPE, 2014.

XVII, 115 p.: il.; 29,7cm.

Orientadores: José Carlos Costa da Silva Pinto

Argimiro Resende Secchi

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Química, 2014.

Referências Bibliográficas: p. 88 – 92.

1. Controle de processos. 2. Segurança de processos.
3. Controle Preditivo Não Linear. I. Pinto, José Carlos Costa da Silva *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

A minha esposa Andréa.

Agradecimentos

Antes de qualquer coisa, gostaria de agradecer ao professor Marcelo D'Oca, pois sem sua ajuda e confiança eu mal teria concluído a graduação.

Agradeço também ao professor Renato Pereira, que me apresentou a área de controle de processos e, conseqüentemente, livrou o Laboratório Kolbe e o professor Marcelo de prejuízos ainda maiores decorrentes de quebras de vidrarias.

À equipe do Laboratório Kolbe, por todo o apoio e pela paciência em esperar a entrega das vidrarias quebradas.

Aos eternos amigos da graduação, que tiveram papel fundamental nos cinco anos que passei em Rio Grande (aos desavisados... sim, este é realmente o nome da cidade!).

Aos novos amigos conquistados nestes dois últimos anos de PEQ, que tornaram tudo mais divertido. Aiany, Alana, Alessandra, Amanda L., Amanda M., Ana, Bruno N., Bruno O., Carla, Cauê, Daniel, Diego, Diogo, Elber, Eliza, Gilliani, Guilherme, Javier, Juliana, Karina, Líniker, Lorena, Luana, Marcela, Maria, Maurício, Mirella, Orsini, Paola, Rayany, Sandra, Tafarel, Thaissa, Tiago, Virgílio... Valeu!

À toda a equipe da G130 (trogloditas do LMSCP), pois sem vocês não haveria “condições hidráulicas” de concluir esta dissertação.

Aos meu orientadores, professor José Carlos e professor Argimiro pelos grandes ensinamentos, pela confiança e por toda a ajuda nesta difícil empreitada chamada “Dissertação do PEQ”.

Agradecimento especial à minha família! Meus pais, tias, avô, prim@s, que me ajudaram demais em todas as etapas da minha vida. Com grande destaque ao apoio dado para cursar a graduação à 2100 km de casa. Valeu a pena!

Mas agradeço sobretudo à minha esposa Andréa, pelo companheirismo, dedicação, paciência e carinho nesses últimos seis anos. Apesar de alguns momentos difíceis, as incontáveis alegrias me fazem ter certeza de que podemos ir muito além desta curta história que começou com passos e conchas. Te amo!

Não posso deixar de agradecer às minha filhas Luna, Nina e Zefa, por seu amor incondicional pelo pai delas!

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

UM SISTEMA INTELIGENTE DE SEGURANÇA DE PROCESSOS QUÍMICOS

Rafael Marinho Soares

Junho/2014

Orientadores: José Carlos Costa da Silva Pinto

Argimiro Resende Secchi

Programa: Engenharia Química

Os sistemas de segurança convencionais são projetados para mitigar incidentes; isto é, têm por objetivo reduzir o impacto de falhas através de válvulas de alívio, discos de ruptura e, em muitos casos, provocam um desligamento de emergência, o chamado *shutdown*. O projeto desses sistemas baseia-se na probabilidade de ocorrência das falhas e na severidade do acidente consequente. Todavia, esta base é obtida empiricamente, deixando o sistema de segurança vulnerável a não linearidades do processo. Para garantir a segurança, as ações de controle são conservadoras, de modo que pequenos desvios aos *setpoints* podem levar ao *shutdown* da unidade, gerando um grande prejuízo econômico. A proposta deste trabalho é simular periodicamente o comportamento do reator diante de falhas em diversos componentes e, a partir desta resposta, determinar o potencial de risco ao qual o sistema está submetido. Dependendo deste potencial, ações preventivas de segurança podem ser tomadas para garantir a segurança e a integridade do reator, evitando, assim, um possível *shutdown*. Estas ações são calculadas de modo a perturbar a planta o mínimo possível, a fim de reduzir o impacto na qualidade do produto, mas manter o processo em operação. O objetivo é aumentar o tempo de operação anual da planta sem comprometer a segurança do processo e a qualidade do produto. Os resultados mostram que a proposta é viável para aplicação em tempo real e, com um modelo confiável e um sistema de detecção de falhas eficiente, é possível evitar *shutdowns* desnecessários.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A SMART SECURITY SYSTEM FOR CHEMICAL PROCESSES

Rafael Marinho Soares

June/2014

Advisors: José Carlos Costa da Silva Pinto

Argimiro Resende Secchi

Department: Chemical Engineering

Conventional safety systems are designed to mitigate incidents, i.e., they aim to reduce the impact of failures using relief valves, rupture disks and, usually, cause a system emergency shutdown. The design of these systems is based on failure likelihood and the accident severity. However, this basis is normally obtained empirically, leaving the safety system vulnerable to process nonlinearities. To ensure the process safety, the control actions are conservative, so that small deviations from setpoints can lead to shutdown, generating large economic losses. This work proposes the periodic simulation of the reactor behaviour considering several component failures and, from these responses, determine the potential risk to which the system is subjected. Depending on this potential, preventive actions can be taken in order to guarantee the reactor safety and integrity and avoid a potential shutdown. These actions are calculated to provoke the least possible disturbance in order to reduce the impact on product quality, while keeping the process operating. The goal is to increase annual operating time of the plant without compromising the safety of the process and product quality. The results show that the proposal is feasible to real time applications and that, with a reliable model and an efficient fault detection system, unnecessary shutdowns can be avoided.

Sumário

Lista de Figuras	x
Lista de Tabelas	xiii
Lista de Símbolos	xiv
Lista de Abreviaturas	xvi
1 Introdução	1
2 Revisão Bibliográfica	3
2.1 O problema de segurança	3
2.2 Projeto de sistemas de segurança	4
2.3 Normas de segurança	8
2.4 Métodos comuns em segurança de processos	11
2.5 Comentários finais	18
3 Método Proposto	19
3.1 O modelo	20
3.2 Métodos numéricos	23
3.3 Controle Preditivo Não Linear	23
3.4 Estimação de estados e parâmetros em tempo real	27
3.5 O sistema inteligente de segurança	29
3.5.1 O simulador de falhas	32
3.5.2 O buscador de estados seguros	32
3.6 Comentários finais	34
4 Resultados e Discussões	36
4.1 Dinâmica do sistema	36
4.1.1 Efeito da temperatura da camisa	37
4.1.2 Efeito da vazão de monômero	37
4.1.3 Efeito da vazão de solvente	38
4.1.4 Efeito da vazão de catalisador	40

4.2	Desempenho do controlador	46
4.3	Desempenho do reconciliador	59
4.4	Simulações de comportamentos inseguros	69
4.4.1	Perda da capacidade de refrigeração do reator	69
4.4.2	Adição de excesso de catalisador	71
4.4.3	Perda de controle da temperatura da camisa	71
4.5	O sistema integrado de segurança	72
4.5.1	Perda da capacidade de refrigeração	73
4.5.2	Adição de excesso de catalisador	75
4.5.3	Perda de controle da temperatura da camisa	77
4.6	Robustez frente a incertezas	80
4.6.1	Incerteza sobre a vazão de catalisador	80
4.6.2	Incerteza sobre a temperatura do reator	83
4.6.3	Considerações finais	83
5	Conclusões	86
5.1	Sugestões para trabalhos futuros	87
	Referências Bibliográficas	88
A	Códigos do MATLAB	93

Lista de Figuras

2.1	Matriz de classificação de risco.	5
3.1	Estrutura do sistema proposto.	19
3.2	Esquema do reator em estudo.	20
3.3	Ilustração do funcionamento do MPC.	25
3.4	Diagrama de blocos do sistema Planta+Reconciliador+Controlador.	28
3.5	Horizonte móvel.	28
3.6	Ilustração da simulação de falhas.	30
3.7	Ilustração do estado seguro.	31
3.8	O sistema integrado de segurança.	31
4.1	Dinâmica do modelo em malha aberta. Temperatura da camisa alta.	38
4.2	Dinâmica do modelo em malha aberta. Temperatura da camisa baixa.	39
4.3	Dinâmica do modelo em malha aberta. Vazão de monômero alta.	40
4.4	Dinâmica do modelo em malha aberta. Vazão de monômero baixa.	41
4.5	Dinâmica do modelo em malha aberta. Vazão de solvente alta.	42
4.6	Dinâmica do modelo em malha aberta. Vazão de solvente baixa.	43
4.7	Dinâmica do modelo em malha aberta. Vazão de catalisador alta.	44
4.8	Dinâmica do modelo em malha aberta. Vazão de catalisador baixa.	45
4.9	Desempenho do controlador SISO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i>	47
4.10	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i> . Primeira sintonia do peso \mathcal{Q} . Variáveis controladas.	49
4.11	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i> . Primeira sintonia do peso \mathcal{Q} . Variáveis manipuladas.	49
4.12	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i> . Segunda sintonia do peso \mathcal{Q} . Variáveis controladas.	50
4.13	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i> . Segunda sintonia do peso \mathcal{Q} . Variáveis manipuladas.	50
4.14	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i> . Terceira sintonia do peso \mathcal{Q} . Variáveis controladas.	52

4.15	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i> . Terceira sintonia do peso \mathcal{Q} . Variáveis manipuladas.	52
4.16	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i> . Primeira sintonia do peso \mathcal{S} . Variáveis controladas.	53
4.17	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 2</i> . Primeira sintonia do peso \mathcal{S} . Variáveis manipuladas.	53
4.18	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 3</i> . Desconsiderando o termo associado ao peso \mathcal{S} . Variáveis controladas. . .	54
4.19	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 3</i> . Desconsiderando o termo associado ao peso \mathcal{S} . Variáveis manipuladas. . .	54
4.20	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 3</i> . Primeira sintonia do peso \mathcal{S} . Variáveis controladas.	56
4.21	Desempenho do controlador MIMO: <i>Setpoint 1</i> \rightarrow <i>Setpoint 3</i> . Primeira sintonia do peso \mathcal{S} . Variáveis manipuladas.	56
4.22	Desempenho do controlador SISO frente ao ruído: <i>Setpoint 1</i> \rightarrow <i>Setpoint 3</i>	57
4.23	Desempenho do controlador MIMO frente ao ruído: <i>Setpoint 1</i> \rightarrow <i>Setpoint 3</i> . Variáveis controladas.	58
4.24	Desempenho do controlador MIMO frente ao ruído: <i>Setpoint 1</i> \rightarrow <i>Setpoint 3</i> . Variáveis manipuladas.	58
4.25	Desempenho do reconciliador frente ao ruído e κ_c constante: $H_{rec} = 5$. 60	
4.26	Desempenho do reconciliador frente ao ruído e κ_c constante: $H_{rec} = 8$. 61	
4.27	Desempenho do reconciliador frente ao ruído e variação degrau no κ_c : $H_{rec} = 5$	62
4.28	Desempenho do reconciliador frente ao ruído e variação degrau no κ_c : $H_{rec} = 8$	63
4.29	Desempenho do reconciliador frente ao ruído e variação exponencial no κ_c : $H_{rec} = 5$	64
4.30	Desempenho do reconciliador frente ao ruído e variação exponencial no κ_c : $H_{rec} = 8$	65
4.31	Desempenho do controlador SISO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,2]$	66
4.32	Desempenho do controlador MIMO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,2]$. Variáveis Controladas.	67
4.33	Desempenho do controlador MIMO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,2]$. Variáveis Manipuladas.	67
4.34	Desempenho do controlador MIMO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,4]$. Variáveis Controladas.	68

4.35	Desempenho do controlador MIMO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,4]$. Variáveis Manipuladas.	68
4.36	Efeito da redução brusca do κ_c	70
4.37	Efeito da redução gradual do κ_c	70
4.38	Perfil da variável manipulada T_c para os cenários de variação do κ_c	70
4.39	Efeito do aumento brusco de F_{in}^{Cat}	71
4.40	Efeito do aumento brusco de T_c	72
4.41	Desempenho do sistema integrado de segurança: $\kappa_c = 0, 1$. Variáveis Controladas e afins.	74
4.42	Desempenho do sistema integrado de segurança: $\kappa_c = 0, 1$. Variáveis Manipuladas.	75
4.43	Desempenho do sistema integrado de segurança: $F_{in}^{Cat} = 1, 5$. Variáveis Controladas e afins.	76
4.44	Desempenho do sistema integrado de segurança: $F_{in}^{Cat} = 1, 5$. Variáveis Manipuladas.	77
4.45	Desempenho do sistema integrado de segurança: $T_c = 375$. Variáveis Manipuladas.	78
4.46	Desempenho do sistema integrado de segurança: $T_c = 375$. Variáveis Controladas e afins.	79
4.47	Robustez do sistema integrado de segurança: falha no medidor de vazão de catalisador. Variáveis Manipuladas.	81
4.48	Robustez do sistema integrado de segurança: falha no medidor de vazão de catalisador. Variáveis Controladas e afins.	82
4.49	Robustez do sistema integrado de segurança: falha no medidor de temperatura do reator. Variáveis Controladas e afins.	84
4.50	Robustez do sistema integrado de segurança: falha no medidor de temperatura do reator. Variáveis Manipuladas.	85

Lista de Tabelas

3.1	Parâmetros do modelo	22
3.2	Limites das variáveis	23
3.3	Configuração assumida pelo controlador em função do risco de operação	26
4.1	Estado estacionário inicial para avaliação da dinâmica do sistema . .	36
4.2	Estados da planta durante as simulações de desempenho do controlador.	47
4.3	Estratégia de seleção dos pesos da função objetivo do controlador . .	48
4.4	Sintonia final do controlador.	59

Lista de Símbolos

C_p	Calor específico da mistura reacional [$\text{J.mol}^{-1}.\text{K}^{-1}$], p. 17
E_a	Energia de ativação [J.mol^{-1}], p. 21
F_{in}	Vazão de alimentação [mol.h^{-1}], p. 20
F_{out}	Vazão de produto [mol.h^{-1}], p. 20
H_{cv}	Horizonte de avaliação de convergência, p. 33
H_{rec}	Horizonte de reconciliação, p. 28
H_{sg}	Horizonte de avaliação do sistema com falha, p. 33
M_i	Quantidade de matéria da espécie i no reator [mol], p. 22
M_t	Quantidade de matéria total do reator [mol], p. 17
M_t	Quantidade de matéria total no reator [mol], p. 20
R	Constante universal dos gases [$\text{J.mol}^{-1}.\text{K}^{-1}$], p. 21
T	Temperatura do reator [K], p. 17
T_c	Temperatura da camisa [K], p. 20
T_{in}	Temperatura da corrente de alimentação [K], p. 20
V	Volume da mistura reacional [m^3], p. 21
X	Conversão da reação química, p. 17
ΔH_r	Entalpia da reação [J.mol^{-1}], p. 17
κ_c	Constante de troca térmica [h^{-1}], p. 20
\tilde{x}	Estado Seguro, p. 33
\hat{y}	Valor medido, p. 28

\mathcal{Q}	Matriz de ponderação do erro de rastreamento das variáveis controladas, p. 25
\mathcal{R}	Matriz de ponderação dos movimentos de controle, p. 25
\mathcal{S}	Matriz de ponderação do erro de rastreamento das variáveis manipuladas, p. 25
\mathcal{X}	Matriz de ponderação do estado seguro, p. 34
θ	Vetor dos parâmetros do modelo dinâmico, p. 27
\mathbf{u}	Vetor das variáveis de entrada do modelo dinâmico, p. 24
\mathbf{x}	Vetor das variáveis de estado do modelo dinâmico, p. 24
\mathbf{y}	Vetor das variáveis de saída do modelo dinâmico, p. 25
k	Constante cinética aparente [mol.h^{-1}], p. 21
k_0	Fator pré exponencial de Arrhenius [$\text{m}^3.\text{mol}^{-1}.\text{h}^{-1}$], p. 21
ω	Falha do processo, p. 33
r	Taxa de reação [mol.h^{-1}], p. 20
t	Tempo [h], p. 20
x	Fração molar, p. 21
M	Horizonte de controle, p. 24
P	Horizonte de predição, p. 24
UA	Coefficiente global de troca térmica [$\text{J.h}^{-1}.\text{K}^{-1}$], p. 21

Lista de Abreviaturas

ANSI	<i>American National Standards Institute</i> , p. 8
BPCS	<i>Basic Process Control System</i> , p. 8
CIA	Agência Central de Inteligência dos EUA, p. 10
CSTR	<i>Continuous Stirred Tank Reactor</i> , p. 34
DASSLC	<i>Differential-Algebraic System Solver in C</i> , p. 23
DDT	Diclorodifeniltricloroetano, p. 10
EKF	<i>Extended Kalman Filter</i> , p. 16
EPI	Equipamento de Proteção Individual, p. 8
EUA	Estados Unidos da América, p. 8
EWDS	<i>Early Warning Detection System</i> , p. 18
HAZOP	<i>Hazard and Operability study</i> , p. 5
IEC	<i>International Electrotechnical Commission</i> , p. 8
ISA	<i>International Society of Automation</i> , p. 8
KGB	Comité de Segurança do Estado da Rússia, p. 10
LMSCP	Laboratório de Modelagem, Simulação e Controle de Processos, p. 2
LOPA	<i>Layer of Protection Analysis</i> , p. 8
MIMO	<i>Multiple Input Multiple Output</i> , p. 26
MPC	<i>Model Predictive Control</i> , p. 23
NLP	<i>Nonlinear Programming</i> , p. 23
NMPC	<i>Nonlinear Model Predictive Control</i> , p. 16

PVI	Problema de Valor Inicial, p. 27
QRA	<i>Quantitative Risk Analysis/Assessment</i> , p. 6
SDE	<i>Shutdown</i> de Emergência, p. 31
SDG	<i>Signed Directed Graphs</i> , p. 13
SIF	<i>Safety Instrumented Function</i> , p. 8
SIL	<i>Safety Integrity Level</i> , p. 14
SISO	<i>Single Input Single Output</i> , p. 26
SIS	<i>Safety Instrumented System</i> , p. 14
SP _{seg}	<i>Setpoint Seguro</i> , p. 31
TCDD	2,3,7,8-tetraclorodibenzo- <i>p</i> -dioxina, p. 9
TNT	Trinitrotolueno, p. 11

Capítulo 1

Introdução

Apesar de a indústria química funcionar em grande escala há dezenas de décadas, somente nas duas últimas o tema segurança ganhou a atenção que merece. Vários acidentes marcaram a história da indústria química, como Chernobyl, Bhopal, Seveso e Sandoz. Lições importantes foram aprendidas com estas tragédias e, dentre elas, destaca-se o conhecimento profundo do processo químico envolvido (MASCHIO *et al.*, 1992 apud CASSON *et al.*, 2012).

Na síntese de um novo processo, mesmo após os testes em escala piloto, o *scale-up* para a escala industrial pode trazer sérias modificações no comportamento dinâmico do sistema. Além destas mudanças, a estrutura de controle pode alterar a estabilidade do sistema em diversos pontos de operação. Neste caso, o uso de modelos matemáticos torna-se indispensável para uma análise dinâmica confiável.

Com o rápido avanço da computação nos últimos anos, o uso de modelos fenomenológicos rigorosos torna-se cada vez mais viável (MANENTI, 2011). Sua penetração no meio industrial, no entanto, ainda é tímida. Nos sistemas de segurança não é diferente. Apesar do ganho evidente que modelos matemáticos não lineares podem trazer em relação à confiabilidade das análises, sua aplicação ainda esbarra nas dificuldades de implementação e hesitação da indústria em substituir os tradicionais modelos lineares. Na área de segurança de processos o cenário é ainda mais árido, visto que, tradicionalmente, as análises são baseadas no conhecimento empírico do processo. Em casos esporádicos, utilizam-se modelos qualitativos, que ainda assim estão bastante defasados em relação às propostas encontradas na literatura científica, que já sugerem a utilização de modelos e técnicas mais precisas (VENKATASUBRAMANIAN *et al.*, 2003).

Além das dificuldades enfrentadas durante o projeto dos sistemas de segurança, novos obstáculos surgem durante a operação industrial. Detectar a origem de eventos indesejados pode ser uma tarefa bastante complexa; porém, uma vez encontrada, pode ser decisiva para determinar a forma de atuação do sistema de segurança (VENKATASUBRAMANIAN *et al.*, 2003). Outro ponto importante é que muitas decisões

que envolvem situações de risco são tomadas pelos operadores, cuja subjetividade reduz consideravelmente a confiabilidade do sistema de segurança. Tudo isso faz com que estes sistemas sejam projetados e operados de maneira conservadora, sob tolerâncias estreitas nos desvios aos *setpoints* e recorrendo muitas vezes ao *shutdown* de emergência como forma de garantir a segurança (LUYBEN, 2012). Estas ações, contudo, levam a consideráveis perdas de produção e, conseqüentemente, perdas financeiras, uma vez que a planta pode ficar ociosa por dias até que esteja preparada para o novo *startup*.

Este trabalho se concentra na operação industrial, colocando o *shutdown* de emergência como última alternativa em situações de risco. Antes dele, é proposta uma etapa de análise dinâmica do sistema, em que se procura uma condição de operação segura para a qual o sistema pode ser deslocado, evitando o *shutdown* e perturbando o sistema o mínimo possível. O objetivo é aumentar o tempo que a planta permanece ativa durante o ano, reduzindo perdas por ociosidade.

Este trabalho é dividido em cinco capítulos. Este primeiro capítulo descreve de forma sucinta o contexto da área de segurança de processos e destaca a abordagem proposta pelos autores. No Capítulo 2, a revisão de literatura descreve os principais tópicos em segurança de processos a serem considerados na contextualização deste trabalho. O Capítulo 3 mostra, em detalhes, a metodologia empregada durante a pesquisa. No Capítulo 4 são apresentados os resultados e validações dos métodos propostos. O Capítulo 5 fecha este trabalho, reunindo as principais conclusões obtidas e algumas sugestões para trabalhos futuros.

Este trabalho foi realizado no Laboratório de Modelagem, Simulação e Controle de Processos (LMSCP) do Programa de Engenharia Química da COPPE, da Universidade Federal do Rio de Janeiro.

Capítulo 2

Revisão Bibliográfica

2.1 O problema de segurança

Segurança é um assunto de extrema importância, sobretudo na indústria química, onde diariamente são processadas toneladas de substâncias perigosas. Para garantir um processo seguro, reduzir impactos ambientais e evitar grandes mudanças posteriores, a segurança deve ser levada em consideração desde a síntese do processo até as práticas diárias de operação (HOWELL, 2009; WOODCOCK & AU, 2013).

As reações exotérmicas fazem dos reatores químicos equipamentos muito perigosos na indústria química. O conhecimento de todas as reações químicas envolvidas no processo é fundamental para o projeto de um sistema seguro de reação. É importante também conhecer a cinética de reações que podem ocorrer em condições anormais do processo, como em temperaturas mais altas, a possibilidade de acúmulo de algum contaminante presente na alimentação, ou ainda, o efeito causado pelo contato entre o fluido da camisa (muitas vezes água, que é uma substância potencialmente reativa) e a massa reacional.

O *runaway* é um dos problemas mais sérios em sistemas exotérmicos (GYGAX, 1988). Este fenômeno é caracterizado pelo aumento autossustentado não controlado da temperatura do reator. Este aumento rápido de temperatura eleva a pressão de vapor dos reagentes no reator e, conseqüentemente, aumenta a pressão no interior do vaso. Além disso, o aumento de temperatura pode desencadear reações secundárias, produzindo mais calor ou gases não condensáveis, que por sua vez causam aumento da pressão no vaso. Neste cenário há um grande risco de vazamento de gases inflamáveis ou mesmo da ruptura das paredes do reator (MANDERS, 2011).

Segundo MASCHIO *et al.* (1992 apud CASSON *et al.*, 2012) a grande maioria dos acidentes ocorridos em sistemas de reação se deveram a:

- ☞ falta de conhecimento acerca da química e da termodinâmica do processo;
- ☞ projeto inadequado do reator (dispositivos de troca térmica, agitação, *scale*

up);

- ∞ sistemas de controle e segurança inadequados;
- ∞ procedimentos de operação e treinamento dos operadores inadequados.

2.2 Projeto de sistemas de segurança

A fim de contextualizar os principais tópicos da área de segurança de processos, o presente trabalho propõe a divisão do projeto de sistemas de segurança em três etapas básicas: (i) identificar os riscos; (ii) determinar a frequência com que as falhas ocorrem e a severidade dessas falhas; e (iii) dimensionar os equipamentos e definir os procedimentos de operação e emergência. A primeira etapa geralmente é qualitativa; ou seja, os riscos, suas causas e consequências são mapeados através de situações hipotéticas e conhecimento empírico do processo. Da mesma forma, é possível mapear a propagação de eventos anormais; entretanto, com o aumento da complexidade do processo, a confiabilidade deste tipo de análise diminui consideravelmente. Nestes casos, técnicas baseadas em modelos matemáticos do processo são recomendadas, mesclando, portanto, a análise qualitativa com elementos quantitativos (GRAF & SCHMIDT-TRAUB, 2001). O resultado da primeira etapa é um mapa com todos os riscos possíveis do processo, incluindo desde problemas pequenos até catástrofes gigantescas. Contudo, estatisticamente, para cada catástrofe, ocorrem milhares de pequenos acidentes; portanto, estes riscos devem ser ponderados de acordo com sua frequência, resultando numa matriz semelhante à da Figura 2.1. Nesta matriz, o risco é o produto entre frequência e severidade, dando origem a três regiões: risco aceitável, risco moderado e risco inaceitável (BARADITS, 2010). Esta matriz é o resultado da segunda etapa do projeto de sistemas de segurança. A terceira e última etapa consiste em tomar medidas para deslocar os eventos críticos (risco inaceitável) para a região de risco aceitável. Esse deslocamento pode seguir uma estratégia horizontal ou vertical. No deslocamento horizontal as medidas têm por objetivo reduzir a frequência dos eventos; já no deslocamento vertical o objetivo é reduzir a severidade dos eventos. Ainda nesta etapa são elaborados os procedimentos de operação e emergência (WOODCOCK & AU, 2013). Os detalhes de cada etapa são descritos a seguir.

Etapa 1 - Identificar os riscos do processo, suas causas e consequências

O método mais utilizado para a identificação dos riscos do processo é o HAZOP (*Hazard and Operability study*), cujos detalhes podem ser encontrados na Seção

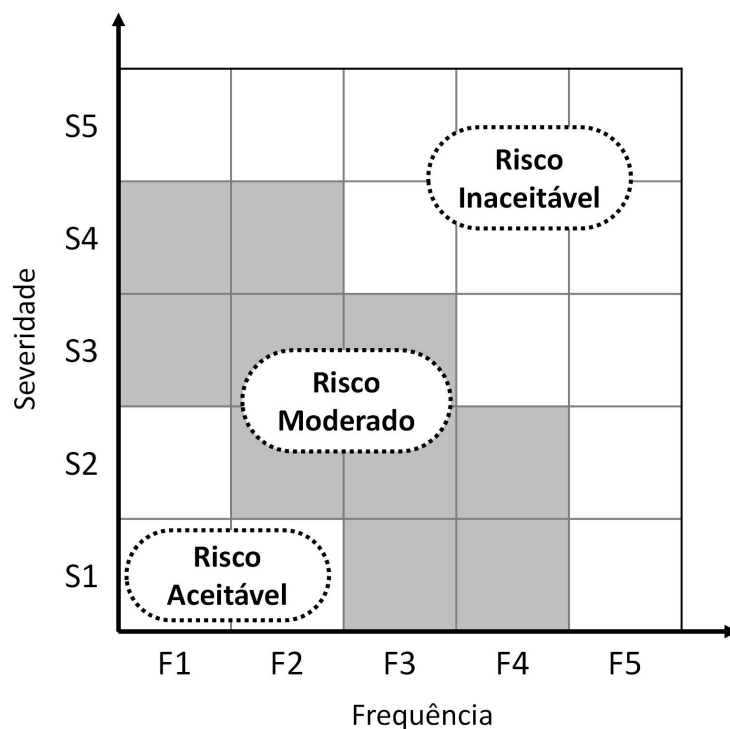


Figura 2.1: Matriz de classificação de risco.

2.4 (VAIDHYANATHAN & VENKATASUBRAMANIAN, 1995; LABOVSKÝ *et al.*, 2007a). A aplicação desse método permite determinar os eventos causadores de risco e suas possíveis consequências de maneira qualitativa. Os sistemas que apresentarem os maiores riscos devem ser mapeados e ser alvo de estudos mais profundos. Esta etapa começa no projeto conceitual da planta; ou seja, a seleção da rota química, por exemplo, pode ser afetada pelo potencial de risco do processo. Uma rota baseada em reagentes tóxicos deve ser evitada por conta do impacto ambiental que um vazamento poderia causar. Da mesma forma, o local onde a planta deve ser construída deve levar em consideração os impactos de contaminação de solo, água e da população residente nos arredores. Se existe uma rota mais segura, esta pode ser escolhida, dependendo, é claro, da diferença de custos de produção. Caso a rota mais perigosa seja escolhida, deve-se atentar para os custos de implantação do sistema de segurança que, naturalmente, serão superiores aos da rota mais segura. Este caso simples mostra a dimensão da importância da segurança nos projetos de processos químicos.

É extremamente importante reunir a maior quantidade possível de informações acerca do processo: informações sobre armazenamento dos reagentes, produtos e catalisadores; as interações que estes produtos podem ter com umidade, temperatura, luz, agitação, etc.; informações sobre o comportamento das reações químicas que podem ocorrer no reator em diferentes condições de temperatura, pressão, agitação e influência de contaminantes presentes na matéria prima. Essas informações po-

dem ser úteis, por exemplo, para determinar a distância que alguns reagentes devem manter de outros reagentes ou equipamentos. Isto ocorre porque a vibração ou o calor podem provocar vazamentos e o contato entre determinadas substâncias pode gerar um incêndio, explosão ou liberação de gases tóxicos. Esse é um cenário típico avaliado nesta etapa de projeto.

Análises do tipo “Árvores de Eventos/Falhas” (*Event/Fault Trees*) também são comuns nesta etapa (VAIDHYANATHAN & VENKATASUBRAMANIAN, 1995; MARHAVILAS *et al.*, 2011). Esses estudos consideram a propagação de falhas e de eventos anormais no sistema. Diversos cenários podem ser criados, incluindo desde pequenos incidentes sem danos até catástrofes com mortes e grandes impactos ambientais. Como as consequências de eventos em sequência podem ser bastantes difíceis de serem previstas com base em conhecimentos empíricos, existem metodologias baseadas em modelos matemáticos do processo capazes de avaliar diversas combinações de cenários. Estas técnicas eram consideradas computacionalmente intensivas e pouco aplicáveis em processos complexos (VENKATASUBRAMANIAN *et al.*, 2003); porém, com o crescente desenvolvimento tecnológico pode constituir uma opção viável.

Etapa 2 - Determinar a probabilidade de ocorrência de eventos anormais

Eventos anormais são difíceis de serem previstos; por isso, deve-se estar sempre preparado para reduzir seus efeitos. Todavia, esses eventos ocorrem com certa frequência e, portanto, podem ser analisados sob o prisma da probabilidade de ocorrência. Este tipo de análise é chamado QRA (*Quantitative Risk Analysis/Assessment*), cujos detalhes podem ser encontrados na Seção 2.4, e é capaz de prover a probabilidade de ocorrência de eventos isolados e possíveis sequências de eventos anormais (usando árvores de falhas). A partir de simulações estocásticas, é possível determinar quais os caminhos mais prováveis de propagação de eventos anormais e qual a probabilidade de ocorrência de cenários catastróficos (PODOFILINI & DANG, 2012). Por legislação, esses cenários devem estar dentro de limites específicos e o QRA é um método indispensável para determinar essas informações (FREDERICKSON, 2002).

Apesar de mais comum no estágio de projeto, o QRA pode ser aplicado *on-line* durante a operação. Isso permite a atualização das probabilidades de falhas de acordo com o histórico do próprio processo e também levando em consideração dados de outras plantas (MEEL & SEIDER, 2008). Aplicações recentes baseadas em modelos do processo são capazes de prever a severidade dos eventos e também podem considerar fatores humanos, como o tempo de resposta frente ao alarme de emergên-

cia (KLOOS & PESCHKE, 2008; DUIJM & GOOSSENS, 2006; HOCHHEIMER *et al.*, 2006). A incorporação de fatores humanos em análises de risco, por sinal, é um tema que tem recebido a atenção de pesquisadores, principalmente porque se estima que 70% dos acidentes são causados por falha humana durante a operação; isto é, erros na tomada de decisão em casos de emergência (VENKATASUBRAMANIAN *et al.*, 2003).

Combinando a primeira e a segunda etapas, obtém-se uma matriz de risco, conforme ilustrado na Figura 2.1. Essa matriz contém todos os eventos anormais possíveis de acontecer no processo, classificados conforme sua frequência (F) e severidade (S). Geralmente, são atribuídas faixas de 1 a 5 para cada característica; portanto, um evento classificado como F1S1 tem baixa frequência e baixa severidade, sendo este o melhor dos cenários. Já um evento classificado como F5S5 (alta frequência e alta severidade) é inadmissível e deve ter seu risco reduzido. A estratégia utilizada para promover esta redução é definida na terceira etapa do projeto de sistemas de segurança (FREDERICKSON, 2002).

Etapa 3 - Projetar os sistemas de segurança e procedimentos de emergência

Com base no mapa de riscos obtido nas etapas anteriores de projeto, é possível definir onde e como atuar para reduzir os riscos e adequá-los à legislação. Um dos métodos utilizados para isso é a LOPA (*Layer of Protection Analysis*), cujos detalhes podem ser encontrados na Seção 2.4. Este método possibilita a criação de camadas de proteção com o objetivo de adequar o risco aos padrões de segurança (FREDERICKSON, 2002). Todos os eventos classificados como “Risco inaceitável” devem ser analisados segundo a LOPA para serem deslocados, pelo menos, até a região de “Risco moderado”. Note que eventos de alta frequência e baixa severidade (F5S1), e vice-versa, não podem ser aceitos. Isto acontece porque os prejuízos de pequenos incidentes ao longo do ano somados podem ser comparados aos causados por incidentes sérios (VENKATASUBRAMANIAN *et al.*, 2003). Portanto, mesmo sendo de baixa severidade, esses eventos devem ter sua frequência reduzida. O mesmo ponto de vista se aplica a eventos de baixa frequência e alta severidade (F1S5). Equipamentos de proteção e medidas de emergência devem ser utilizados para reduzir os impactos em cenários graves, ainda que a probabilidade destes eventos seja pequena. De uma maneira geral, os níveis de proteção empregados no processo são (FREDERICKSON, 2002): o projeto do processo em si, o sistema básico de controle (BPCS - *Basic Process Control System*), alarmes e ações dos operadores, válvulas de alívio e, por fim, as *Safety Instrumented Function* (SIF). Cada uma das camadas tem uma confiabilidade finita, ou seja, uma probabilidade de falha conhecida. A

probabilidade de falha acumulada é o produto entre a probabilidade do evento e as probabilidades de falha das camadas. A quantidade de camadas depende do nível de segurança necessário para o sistema.

Outras medidas importantes são os protocolos de segurança e emergência (HOWELL, 2009; WOODCOCK & AU, 2013). O primeiro tem relação com o comportamento dos funcionários durante a operação (*Safety culture*); por exemplo, isolar a área próxima ao reator enquanto o mesmo estiver em funcionamento, fazer uso contínuo de EPI (Equipamento de Proteção Individual), aplicar as políticas de gestão de mudanças, etc. O segundo tem foco no comportamento dos funcionários durante situações de emergência, como resposta aos alarmes, combate ao incêndio, evacuação da planta, etc. Todos esses aspectos devem ser bem treinados, pois estima-se que funcionários sob estresse têm até 100 vezes mais chances de tomar decisões erradas do que funcionários sem estresse (FREDERICKSON, 2002).

Depois de grandes desafios durante o projeto dos sistemas de segurança, a operação da planta traz um novo desafio: identificar as falhas. Durante o projeto, foi assegurado que o sistema está preparado para lidar com qualquer problema, desde que mudanças em certos parâmetros sejam reconhecidas. Em diversas situações, reconhecer esses desvios não é tarefa simples, e um problema no reconhecimento pode comprometer a segurança do processo. Assim, a detecção de falhas é um tema que tem recebido bastante atenção de pesquisadores (VENKATASUBRAMANIAN *et al.*, 2003).

2.3 Normas de segurança

Apesar da importância do projeto e gerenciamento de sistemas de segurança na indústria química, o desenvolvimento de normas e padrões de referência surgiu aos poucos, conforme a demanda, geralmente motivado por acidentes trágicos. A primeira tentativa de padronizar o gerenciamento de segurança na indústria química foi a Diretiva de Seveso (BARADITS, 2010). Criada após um grave acidente em Seveso, na Itália, sofreu alterações ao longo dos anos após outros acidentes marcantes, fazendo deste um excelente exemplo da evolução do gerenciamento de segurança na indústria (mais detalhes na Seção *Diretivas de Seveso*). Nos EUA, a primeira norma criada foi a ANSI/ISA 84.01-1996 (ISA, 1996). Outra iniciativa europeia foi a IEC 61508 (IEC, 1999), voltada para o setor industrial em geral, e atualizada em 2003 para a atual IEC 61511 (IEC, 2003), cujo escopo foi ampliado para a indústria química, óleo e gás, petroquímica e farmacêutica. Nos EUA, a IEC 61511 foi aplicada através da ANSI/ISA 84.00.01-2004 (ISA, 2004) e é considerada pela indústria como *Boas Práticas de Engenharia*. Na Europa, o cumprimento de Diretivas é obrigatório, enquanto as normas (*Standards*) são apenas recomendadas (BARADITS, 2010).

Diretivas de Seveso

A Diretiva de Seveso é a principal norma da legislação da União Europeia referente ao controle do risco de acidentes graves envolvendo substâncias perigosas. A norma obriga empresas que utilizam produtos perigosos a identificar e controlar os riscos de seus processos, bem como desenvolver planos de emergência para lidar com possíveis acidentes. Esta diretiva foi criada após um grave acidente em Seveso, na Itália. Com o passar do tempo, outros acidentes motivaram atualizações na norma, dando origem às chamadas “Seveso II” e “Seveso III”.

O acidente de Seveso ocorreu em 1976 em uma planta química responsável pela fabricação de pesticidas e herbicidas. Um aumento na temperatura do reator acelerou duas reações exotérmicas, que causaram um *runaway* e a síntese do subproduto tóxico e carcinogênico TCDD (2,3,7,8-tetraclorodibenzo-*p*-dioxina). Após um vazamento, uma nuvem de vapor do produto se espalhou pela região, contaminando dezenas de quilômetros quadrados de solo e vegetação. Apesar de não ter causado mortes imediatas, houve liberação de alguns quilogramas da substância que, mesmo em quantidades da escala de microgramas, pode ser letal para o homem. Mais de 600 pessoas foram evacuadas de suas casas e cerca de 2000 receberam tratamento para desintoxicação (EUROPEAN COMMISSION, 2014). Milhares de animais foram sacrificados para evitar que a dioxina entrasse na cadeia alimentar. Depois do acidente, o gerente da fábrica e os autores da tecnologia de processo afirmaram que não tinham conhecimento de que estas reações paralelas poderiam ocorrer no meio reacional (MOLNÁR, 2005). Após este grave acidente, a União Europeia decidiu adotar uma legislação com o objetivo de prevenir e controlar acidentes em indústrias químicas, a chamada “Diretiva de Seveso” ou “Seveso I”. Adotada em 1982, a diretiva obrigava as indústrias a terem um procedimento padrão de comportamento em casos de acidente, como sistemas de segurança, plano para mitigação de danos à planta e à vizinhança e evacuação da planta.

A Diretiva de Seveso recebeu dois adendos ainda na década de 80, em virtude dos graves acidentes em Bhopal, na Índia, e em Basel, na Suíça. O primeiro, considerado o maior desastre de todos os tempos na indústria química, aconteceu em 1984 em uma fábrica do pesticida DDT (diclorodifeniltricloroetano) da empresa americana Union Carbide. Um vazamento de 32 toneladas de metil-isocianato, precursor do DDT, atingiu milhares de moradores das redondezas da planta. O número de mortos é bastante controverso; porém, estima-se que milhares de pessoas morreram por ação direta da substância e outras centenas de milhares sofrem até hoje devido à contaminação (ECKERMAN, 2005). A empresa foi acusada de não oferecer os níveis mínimos de segurança para este tipo de instalação, como manutenção adequada e treinamento dos operadores para manuseio de substâncias perigosas. O

acidente ocorreu por conta da adição de água no tanque que armazenava o metil-isocianato, causando um *runaway*. O conseqüente aumento da pressão no tanque levou à abertura da válvula de alívio e, então, à liberação de uma nuvem de vapor tóxico (ECKERMAN, 2005). Falha mecânica, falha de operação, sabotagem... Várias foram as teorias acerca do motivo para a entrada de água no tanque, mas a verdade ainda é um mistério.

Em 1986, na cidade de Basel, na Suíça, após um incêndio no depósito da indústria agroquímica Sandoz, a água de combate a chamas foi contaminada com mercúrio, pesticidas à base de organofosfatos, organoclorados e outras substâncias perigosas armazenadas no depósito, resultando na descarga de toneladas de poluentes no rio Reno. A contaminação conferiu coloração vermelha às águas do rio e foi responsável pela morte de meio milhão de peixes (EUROPEAN COMMISSION, 2014). A causa do incêndio nunca foi descoberta, mas em 2000, um oficial da agência de inteligência norte-americana CIA acusou a agência de segurança russa KGB de sabotagem. Segundo ele, o objetivo seria desviar a atenção do desastre nuclear de Chernobyl, ocorrido seis meses antes na Ucrânia (SWISSINFO, 2000).

O acidente nuclear de Chernobyl ocorreu em abril de 1986. O reator, assim como outros, tinha um sistema de emergência para geração de energia elétrica através de geradores à diesel. Caso a fonte principal de energia (o próprio reator nuclear) fosse desligada, a planta teria energia suficiente para manter o sistema sob controle. Entretanto, esses geradores levavam trinta segundos para ligar, tempo considerado muito longo para religar as bombas e restabelecer a circulação da água de refrigeração do reator. Decidiu-se então adicionar uma terceira fonte de energia, direcionando a produção das turbinas, que reduzem a velocidade em caso de emergência, para as bombas de refrigeração até que os geradores ligassem. No início dos testes, os operadores desligaram diversos dispositivos responsáveis pelo *shutdown* automático do reator, com o objetivo de promover livremente vários testes com a nova fonte de energia. Durante os testes, houve um aumento inesperado da temperatura do reator, provocando a vaporização do fluido de refrigeração. As tentativas manuais de *shutdown* não foram efetivas e em segundos as paredes de metal em torno do bloco de grafite e a espessa camada de concreto no topo do reator explodiram. Um teste do sistema de segurança, portanto, causou o pior acidente nuclear da história (SAGAN, 1993, p. 161).

Em 1996 foi adotada a “Diretiva de Seveso II” ou “Seveso II”, com revisões e atualizações acerca do escopo da legislação, do gerenciamento de sistemas de segurança, plano de emergência e uso de terras (EUROPEAN COMMISSION, 2014). Este último se refere à adequação da distância entre estabelecimentos de processamento ou estocagem de produtos perigosos e áreas públicas, residências, vias de transporte e áreas de proteção ambiental. Por causa dos acidentes de Toulouse, Baía Mare

e Enschede, esta diretiva recebeu algumas alterações em 2003, principalmente nas áreas de mineração, processamento de substâncias explosivas e pirotécnicas e armazenamento de nitrato de amônio e seus derivados, como fertilizantes (EUROPEAN COMMISSION, 2014).

O acidente de Toulouse, na França, ocorreu em 2001. Devido a falhas no armazenamento de nitrato de amônio, cerca de 40-80 toneladas desta substância foram consumidas numa explosão equivalente a 20-40 toneladas de TNT (referência para cálculos de eventos pirotécnicos). A explosão causou a morte de 30 pessoas e mais de 2500 feridos. As causas ainda são desconhecidas. Uma vez que para detonação do produto é necessária uma fonte de energia, as condições de armazenamento eram consideradas seguras (LAURENT, 2001).

O maior desastre ecológico depois de Chernobyl, foi como ficou conhecido o acidente na empresa de mineração de ouro Aurul, em Baia Mare, na Romênia, em 2000 (MERGEL, 2011). Mais de 100 mil metros cúbicos de água contaminada com cianeto (estimado em 100 toneladas) vazaram de uma represa, contaminando os rios Săsar, Szamos, Tisza e Danúbio, matando uma grande quantidade de peixes. O vazamento foi resultado de falhas do projeto da represa, que não foi capaz de suportar o excesso de água proveniente de intensas chuvas e do derretimento de neve na região. Apesar de ser uma rota bastante perigosa, o cianeto era usado como agente complexante. A ligação entre cianeto e ouro forma um complexo solúvel em água, viabilizando a exploração de terras com baixos teores do mineral (MERGEL, 2011).

A explosão de mais de 100 toneladas de fogos de artifício em uma região pobre de Enschede, na Holanda, em 2000, é considerado o maior acidente da história envolvendo este material. A explosão provocou a morte de 26 pessoas, dentre elas, 4 bombeiros. Mais de 400 casas foram destruídas e 947 pessoas ficaram feridas. Houve também dispersão de amianto na atmosfera (VOOGD, 2004).

Em 2012, foi publicada a “Diretiva de Seveso III” ou “Seveso III”, que atualiza a classificação de produtos químicos, melhora o acesso público ao conteúdo da diretiva e torna mais rígido o padrão de fiscalização dos estabelecimentos, para garantir o cumprimento das normas de segurança. O prazo final para total adaptação à Seveso III é junho de 2015 (EUROPEAN COMMISSION, 2014).

2.4 Métodos comuns em segurança de processos

HAZOP

Na indústria química, as causas e consequências de eventos indesejados podem ser mapeadas qualitativamente utilizando a técnica HAZOP (*Hazard and Operabi-*

lity study). Esta técnica surgiu na década de 60, mas sua primeira publicação na literatura científica ocorreu em 1974 (LAWLEY, 1974). A técnica tem por objetivo identificar riscos (causas e consequências) e projetar ações de emergência baseado na configuração e operação da planta. Alguns termos de orientação típicos utilizados durante a análise são: nada, mais, menos, parte de, ao invés de, cedo, tarde, antes, depois e outros. Estes termos, combinados com variáveis e parâmetros do processo, formam cenários do tipo: temperatura alta/baixa, agitação rápida/lenta, nível alto/baixo/inexistente, reação química rápida/lenta/inexistente/indesejável.

Por causa desse caráter subjetivo, várias situações de risco podem ser negligenciadas ou subestimadas, comprometendo o desempenho do sistema de segurança (GRAF & SCHMIDT-TRAUB, 2001). Além disso, o HAZOP é um método que exige bastante tempo e envolve muitos profissionais durante sua execução. Assim, o desenvolvimento de técnicas computacionais tornou-se uma necessidade, principalmente em processos de alta complexidade. As principais vantagens desta abordagem são (LABOVSKÝ *et al.*, 2007a): grande redução de tempo e mão de obra; estudo mais rigoroso e detalhado; redução da influência de fatores humanos. Geralmente, o HAZOP não leva em consideração informações como duração da falha, ou quanto tempo levará para que o sistema viole os limites de segurança, ou mesmo a amplitude das consequências causadas pela falha. De uma maneira geral, o HAZOP é executado com auxílio de tabelas e conhecimento empírico, sendo raramente utilizados modelos matemáticos dinâmicos.

Com o objetivo de automatizar o procedimento HAZOP, VAIDHYANATHAN & VENKATASUBRAMANIAN (1995) propuseram o uso de modelos baseados em dígrafos, que são representações de vértices conectados por arestas orientadas, ou seja, a conexão ocorre de V1 para V2, mas não de V2 para V1. O HAZOP baseado em dígrafos foi testado industrialmente e foi capaz de identificar quase todos os riscos previstos pela equipe tradicional de HAZOP; porém, em muito menos tempo. O uso de dígrafos para construção de árvores de eventos continua em desenvolvimento. Alguns pesquisadores propõem também o uso de *Signed Directed Graphs* (SDG), que são dígrafos em que cada aresta tem um sinal positivo ou negativo. Sinais positivos indicam que as variáveis conectadas têm a mesma tendência de comportamento, enquanto sinais negativos indicam tendências opostas. Além de poder ser utilizada durante o projeto, esta técnica pode ser aplicada durante a operação para detecção de falhas, pois é possível realizar análises tanto no sentido causa-efeito quanto no sentido efeito-causa (VENKATASUBRAMANIAN *et al.*, 2000; MAURYA *et al.*, 2007; WANG *et al.*, 2009). As técnicas baseadas em dígrafos descrevem muito bem processos contínuos; entretanto, têm sérias limitações para análises de processos dinâmicos, como bateladas. Assim, alguns pesquisadores propõem a utilização conjunta de dígrafos e redes de Petri, sendo estas mais adequadas para análises

dinâmicas, porém com limitações para processos contínuos (SRINIVASAN & VENKATASUBRAMANIAN, 1996; WANG, 2002).

LABOVSKÝ *et al.* (2007a; 2007b) propuseram a utilização de modelos matemáticos para a identificação de riscos; ou seja, um HAZOP baseado em modelo. Seus resultados mostraram que, como esperado, a acurácia do modelo influencia a identificação de riscos. Portanto, apesar de o HAZOP baseado em modelo ser muito superior ao tradicional, sua confiabilidade está condicionada à invariância dos parâmetros do modelo. Caso esses parâmetros mudem ao longo da operação, o comportamento dinâmico do sistema é alterado e, conseqüentemente, os riscos também podem sofrer modificações. A abordagem proposta pelos autores leva em consideração apenas um modelo autônomo; isto é, não avalia a influência da malha de controle nem a influência de ruído no sinal da variável medida.

QRA

O objetivo do QRA (*Quantitative Risk Analysis/Assessment*) é atribuir um valor numérico ao risco, que é definido como a probabilidade associada a algum tipo de dano ou perda (HAWKINS, 2014). O principal tipo de perda a ser evitado é a morte; por isso, o risco em indústrias químicas é medido em termos de *fatalidades por ano*. Após o mapeamento de todos os equipamentos que podem apresentar algum tipo de risco, são utilizados bancos de dados com informações sobre frequências de falhas desses equipamentos. Esses bancos de dados são construídos a partir do histórico de dados do processo. Com essas informações, é possível criar combinações de falhas que conduzem a diversos cenários de risco (PASMAN, 2009; PODOFILLINI & DANG, 2012). Cada cenário consiste da localização do equipamento, da magnitude da falha, da temperatura, pressão e composição do material que poderia vazar em virtude da falha e a frequência com que a falha deve ocorrer. Este tipo de método é aplicado principalmente para casos de dispersão de vapores explosivos e tóxicos na atmosfera que, através de modelos matemáticos, têm sua área de dano estimada, servindo de base para a definição da distância entre a planta e áreas urbanas. O QRA é utilizado também para definição de planos de mitigação de riscos e análises de custo-benefício para implantação de esquemas de redução de risco, como tubulações de maior espessura, bombas de selo duplo, etc (HAWKINS, 2014).

MARSEGUERRA & ZIO (1996) propuseram a utilização de simulações de Monte Carlo para cálculos de propagação de falhas. Os autores recomendaram esta abordagem para analisar simultaneamente a evolução do sistema dinâmico e eventos anormais em equipamentos. Este tipo de abordagem ficou conhecido como QRA dinâmico. Apesar da maior confiabilidade do QRA dinâmico, em relação ao QRA tradicional, essa abordagem não teve grande penetração no meio industrial devido

às dificuldades de implantação (LABEAU *et al.*, 2000). A incorporação da dinâmica do processo nos estudos de propagação de falhas começou na década de 80, mas o tema ainda é bastante pesquisado.

MEEL & SEIDER (2008) propuseram um método de análise quantitativa de risco em sistemas de segurança em tempo real baseado em estatística Bayesiana e funções cópulas. Tal metodologia é capaz de prever: (1) a frequência de ocorrência de eventos anormais, (2) a confiabilidade de equipamentos e ações humanas e (3) a propagação de eventos anormais nos sistemas de segurança. As probabilidades de falhas são atualizadas em tempo real através da análise Bayesiana e, através de cópulas, levam em consideração informações de desempenho do sistema de segurança local, de outras unidades da planta e, ainda, de outras plantas.

PODOFILLINI & DANG (2012) avaliaram a influência da definição das probabilidades de sucesso/fracasso, empregadas em árvores de eventos para estudos de propagação de falhas, no resultado do QRA tradicional e compararam com o resultado do QRA dinâmico (utilizando simulações de Monte Carlo). Segundo os autores, uma vez que estas probabilidades estão corretamente definidas, não há diferenças entre as abordagens tradicional e dinâmica. Contudo, por ser mais conservadora, é recomendável a utilização da técnica tradicional quando as probabilidades não possuem grande confiabilidade.

LOPA

O LOPA (*Layer of Protection Analysis*) é uma técnica de análise de riscos de processos baseado nas informações obtidas durante o HAZOP e o QRA; ou seja, severidade, causas e probabilidade de ocorrência de eventos perigosos (FREDERICKSON, 2002). Utilizando os padrões de risco da empresa, é possível analisar a redução de risco obtida para várias camadas de proteção. Mesmo após o projeto do processo, do sistema de controle básico, dos alarmes e respectivas ações dos operadores e das válvulas de alívio, uma redução adicional de risco pode ser necessária. Neste caso, uma SIF (*Safety Instrumented Function*) deve ser instalada. Através do LOPA, é possível calcular a confiabilidade necessária à SIF para atingir os níveis exigidos de segurança. Esta confiabilidade é chamada SIL (*Safety Integrity Level*). SIFs são malhas instrumentadas que atuam sobre algum risco específico do processo. Em outras palavras, uma SIF é simplesmente a lógica aplicada para alcançar uma determinada redução de risco. O *hardware* e o *software* usados para aplicar uma SIF são chamados de SIS (*Safety Instrumented System*)(SUMMERS, 2003).

Detecção de falhas

Falhas são desvios indesejados das variáveis em relação a uma faixa aceitável. Segundo VENKATASUBRAMANIAN *et al.* (2003), esses eventos podem ser classificados em três tipos, como definidos a seguir.

a - Mudanças grosseiras de parâmetros do modelo

Mudanças grosseiras são mudanças em variáveis independentes do processo; isto é, variáveis que não apresentam um padrão definido de comportamento e, por isso, não possuem um modelo dinâmico. Exemplos dessas falhas são mudanças indesejadas na concentração de um reagente na corrente de alimentação ou, ainda, mudanças no coeficiente de troca térmica devido à incrustação no trocador de calor. Vale ressaltar que, no último caso, o uso de um modelo dinâmico é possível se a taxa de deposição de material nas paredes do trocador for conhecida.

b – Mudanças estruturais

Mudanças estruturais são mudanças no processo em si, de forma que o modelo matemático precise ser alterado para recuperar a representatividade do processo. Exemplos são falhas do controlador, válvulas emperradas e vazamentos em tubulações (VENKATASUBRAMANIAN *et al.*, 2003).

c – Mau funcionamento de sensores e atuadores

Mau funcionamento são falhas em instrumentos de medida e elementos finais do sistema de controle, como válvulas. Essas falhas podem provocar *bias* constante no sinal de controle ou indicar medidas errôneas que, por sua vez, podem causar instabilidade no controlador. Por conta do ruído frequentemente presente nas medidas, detectar essas falhas pode constituir um grande desafio. Por isso, a instrumentação deve estar sempre calibrada e, sempre que possível, apresentar redundância (possibilidade de duas ou mais inferências simultâneas da variável).

Detectar falhas não é uma tarefa simples. Depois de tomadas todas as medidas especificadas no projeto dos sistemas de segurança e aplicação das rotinas de operação segura, um dos desafios passa a ser detectar as falhas (SERPAS, 2013). De uma maneira geral, os métodos para detecção de falhas se dividem em duas categorias (VENKATASUBRAMANIAN *et al.*, 2003): métodos baseados em modelo de processo e métodos baseados em dados de processo. Os métodos baseados em modelo têm recebido atenção da indústria devido às características sistemáticas de execução. Para modelos quantitativos destacam-se: observadores de diagnóstico, relações de paridade, filtro de Kalman e estimação de parâmetros. Para modelos qualitativos destacam-se: grafo dirigido com sinal (*signed directed graph*), árvore de falhas,

simulação qualitativa e teoria qualitativa de processo. Na ausência de modelos matemáticos confiáveis, o estudo é realizado com base no histórico de dados do processo, cujas principais técnicas são: sistemas especialistas (*expert systems*) e análise qualitativa de tendências (ambas qualitativas); redes neurais, análise de componentes principais e classificadores estatísticos (técnicas quantitativas). Detalhes sobre as técnicas podem ser encontrados no trabalho de VENKATASUBRAMANIAN *et al.* (2003).

DESHPANDE *et al.* (2007) propuseram uma metodologia integrada entre NMPC (*Nonlinear Model Predictive Control*) e detecção de falhas, em que o modelo do controlador é atualizado sempre que um desvio sistemático em algum parâmetro é identificado. Esta identificação é realizada através da comparação entre o valor do sinal recebido e sua distribuição de probabilidades, previamente conhecida. Quando um desvio é identificado, sua magnitude é estimada pelo Filtro de Kalman Estendido (*Extended Kalman Filter* - EKF), assim como o momento em que a falha ocorreu. O modelo é então atualizado, fazendo com que o controlador se adapte para manter as variáveis no *setpoint*. Devido à dificuldade em distinguir uma falha de uma mudança de ponto de operação, esta abordagem é bem sucedida quando as condições de operação da unidade são bem conhecidas.

REIS (2008) propôs diversas estruturas de controle tolerante a falhas reunindo diferentes técnicas de identificação e diagnóstico de falhas baseadas no histórico de dados e em modelos de processo. Segundo o autor, cada processo requer uma estrutura de controle tolerante específica, por isso o conhecimento das características das principais técnicas de detecção de falhas é importante.

Um problema recorrente no meio industrial é o excesso de ativação de alarmes. Por falha de projeto ou de atualização de sintonia, alguns alarmes permanecem ativados mesmo durante uma operação normal, dificultando a percepção de desvios indesejados das variáveis. Por isso, o gerenciamento de alarmes ganhou bastante atenção de pesquisadores e engenheiros de processos recentemente. Como resultado, foram publicadas as normas EEMUA (2007) e ANSI/ISA 18.2-2009 (ISA, 2009), que regulam o projeto, implantação e operação de sistemas de alarmes.

Controle de *runaway*

Falha na dosagem de reagentes e catalisadores, presença ou adição de contaminantes e perda de controle da temperatura são os principais eventos que podem levar ao *runaway*. As principais abordagens ligadas ao controle deste fenômeno são: ventilação, contenção, ventilação e contenção e inibição da reação (DAKSHINAMOORTHY *et al.*, 2004). Nas três primeiras abordagens, a preocupação está no dimensionamento correto de válvulas de alívio e discos de ruptura, que requerem

conhecimento da cinética de reação durante o *runaway* (MANDERS *et al.*, 2011). Quando se trata de um reator batelada de único produto, esta pode ser uma tarefa relativamente simples. Entretanto, em reatores multiproduto e sistemas contínuos, a quantidade de variáveis que podem afetar o comportamento do sistema é grande e, por isso, uma modelagem matemática rigorosa torna-se indispensável. Mas o grande problema daquelas abordagens é que, mesmo funcionando apropriadamente, o conteúdo do reator deve ser direcionado para um local seguro. Caso contrário, pode ocorrer a formação de uma nuvem explosiva, que poderia sofrer ignição e causar sérios danos (MANDERS *et al.*, 2011). Portanto, apesar de indispensáveis, os sistemas de alívio devem ser acionados apenas como último recurso. Assim, o estudo de agentes inibidores ganhou a atenção dos pesquisadores.

Inibidores são basicamente classificados em dois tipos: diluentes frios e inibidores químicos. A adição de diluentes frios provoca a diminuição da temperatura do reator e, conseqüentemente, da taxa de reação durante o *runaway*. Entretanto, a adição de um diluente pode provocar grandes perturbações no processo e comprometer o sistema. Os inibidores químicos reagem com os reagentes e/ou catalisadores, competindo com a reação de *runaway*. Esta abordagem tem se mostrado bastante eficiente no controle deste fenômeno (AMPELLI *et al.*, 2006; DAKSHINAMOORTHY *et al.*, 2004).

Um método bastante interessante e proposto por MANDERS *et al.* (2011) para evitar o *runaway* em processos de polimerização é manter a quantidade de monômero não reagido no reator abaixo de um limite predeterminado. Esta proposta é baseada no fato de que, em caso de *runaway*, altas conversões são alcançadas, ou seja, o reagente limitante é totalmente consumido. Como o aumento de temperatura é rápido, o processo é quase adiabático, pois a capacidade de remoção de calor do sistema de refrigeração é muito inferior à taxa de calor produzida pela reação química. Pode-se, portanto, calcular a temperatura máxima do reator através da Equação 2.1, que resulta na temperatura adiabática para conversão total do monômero. Assim, a temperatura máxima possível em caso de *runaway* pode ser conhecida; conseqüentemente, a pressão máxima também pode ser conhecida e, portanto, a quantidade máxima de monômero segura no reator é função da pressão máxima admitida pelo vaso.

$$T_{adiabatica} = \frac{(1 - X)(-\Delta H_r)}{M_t C_p} + T \quad (2.1)$$

Em que X é a conversão, ΔH_r é a entalpia da reação, M_t é a massa total no reator, C_p é o calor específico da mistura reacional e T é a temperatura do reator.

Seguindo uma linha de raciocínio preventiva, STROZZI & ZALDÍVAR (1994) propuseram uma técnica de detecção antecipada de *runaway*, que se mostrou bas-

tante eficaz para detecção deste fenômeno. Esse tipo de método é chamado de *Early Warning Detection System* (EWDS) e já foi amplamente estudado por diversos pesquisadores; contudo, a abordagem dos referidos autores tem ganhado destaque na literatura (ZALDÍVAR *et al.*, 2003). A técnica se baseia na caracterização da concavidade da trajetória dinâmica do processo.

2.5 Comentários finais

Neste capítulo, foram apresentados os principais métodos utilizados pela indústria para lidar com o problema de segurança, bem como as principais normas que a regulam. Viu-se que, ao longo da história, a importância dada para a segurança no meio industrial foi motivada por sérios acidentes, que poderiam ter sido evitados se políticas mais rígidas de projeto e operação fossem seguidas. De modo geral, a segurança industrial ainda é tratada de forma bastante empírica; isto é, baseado em experiências de processos anteriores ou similares. Apesar da alta complexidade dos sistemas químicos, aplicações envolvendo o uso de modelos matemáticos não são muito comuns, sobretudo, tratando-se de modelos não lineares. Por isso, muitas decisões são deixadas por conta do operador que, como qualquer outro ser humano, está sujeito a falhas.

Os sistemas de polimerização são exemplos clássicos em estudos de dinâmica não linear por sua natureza altamente exotérmica e por apresentarem fenômenos singulares como o efeito gel, tornando o uso de modelos fenomenológicos imprescindível para a representação confiável do processo. Contudo, os atuais esquemas de segurança não contemplam este tipo de modelo, criando uma lacuna tecnológica que pode começar a ser preenchida através do presente trabalho.

Capítulo 3

Método Proposto

Este capítulo descreve todas as técnicas utilizadas no desenvolvimento deste trabalho com o objetivo de possibilitar a completa reprodução de seus resultados.

A estrutura do sistema perseguido neste trabalho é ilustrada na Figura 3.1. As variáveis medidas do processo são enviadas ao reconciliador de dados, que estima os valores das variáveis de estado e os valores dos parâmetros do modelo matemático do processo. Os dados reconciliados são enviados para o controlador, que calcula as novas ações de controle para atingir ou manter as variáveis controladas em seus *setpoints*. Periodicamente, os dados reconciliados são enviados também para o sistema inteligente de segurança, para simulações de comportamentos inseguros e avaliação do risco ao qual o processo está sujeito. Dependendo do resultado da avaliação do risco, o sistema de segurança pode calcular novos *setpoints* para garantir a segurança do processo.

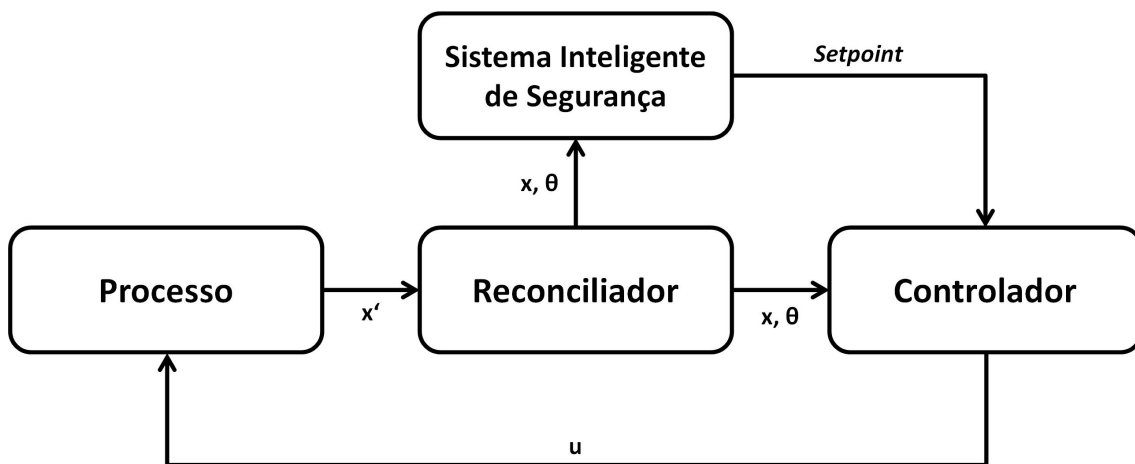


Figura 3.1: Estrutura do sistema proposto.

Os detalhes acerca do processo, do modelo matemático e de seus parâmetros são descritos na Seção 3.1. Os métodos numéricos utilizados nas simulações são definidos na Seção 3.2. A estratégia de controle é detalhada na Seção 3.3. A Seção 3.4 aborda

a metodologia empregada para a estimação de estados e parâmetros do processo. O funcionamento do sistema inteligente de segurança é descrito detalhadamente na Seção 3.5. E este capítulo se encerra com uma breve discussão na Seção 3.6.

3.1 O modelo

A fim de ilustrar a ideia proposta nesta dissertação, considere o esquema mostrado na Figura 3.2. O sistema consiste em um reator tanque agitado com camisa, alimentado por três correntes de entrada (monômero, solvente e catalisador) e com uma corrente de saída (produto). A reação química considerada simula uma polimerização em suspensão, em que o monômero reage para formar polímero na presença de um solvente (ou diluente) e de um catalisador. Monômero, solvente, polímero e catalisador estão na fase líquida.

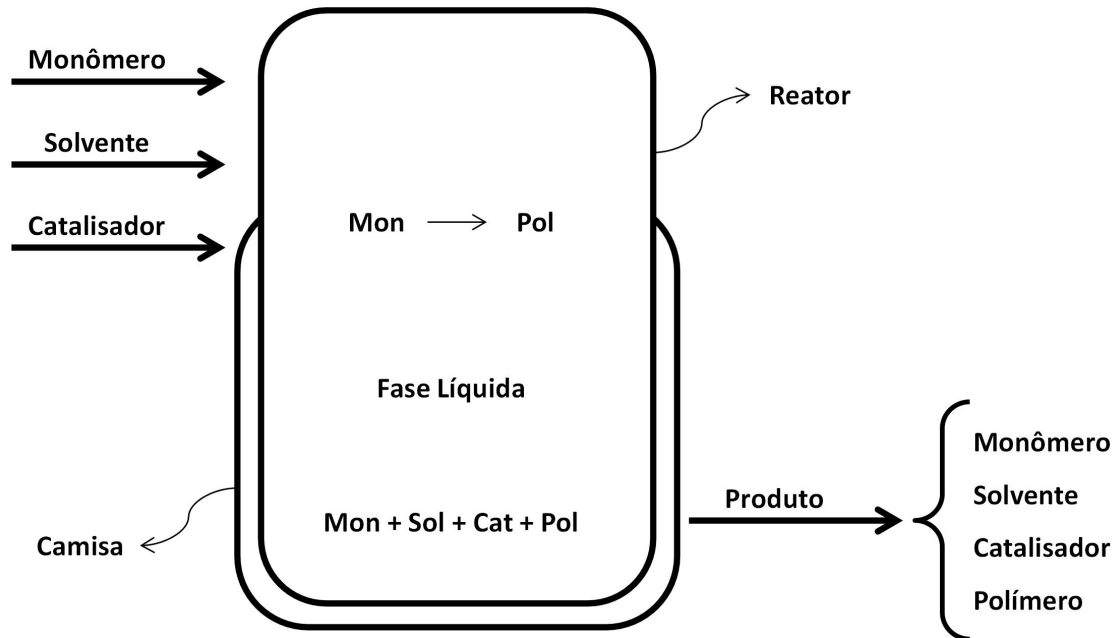


Figura 3.2: Esquema do reator em estudo.

O modelo matemático deste sistema é composto por cinco equações, sendo uma referente ao balanço de energia (Equação 3.1) e as demais referentes aos balanços de massa do monômero, polímero, solvente e catalisador (Equações 3.3 - 3.6, respectivamente).

O balanço de energia do reator é dado por:

$$\frac{dT}{dt} = \frac{F_{in} T_{in}}{M_t} - \frac{F_{out} T}{M_t} + \frac{\Delta H_r r}{M_t C_p} + \kappa_c(T_c - T) \quad (3.1)$$

Em que a constante de troca térmica κ_c é dada pela equação:

$$\kappa_c = \frac{UA}{M_t C_p} \quad (3.2)$$

Os balanços de material foram realizados em termos de quantidade de matéria. O balanço de monômero no reator, é dado por:

$$\frac{dM_{Mon}}{dt} = F_{in}^{Mon} - F_{out}^{Mon} - r \quad (3.3)$$

O balanço de polímero no reator é dado por:

$$\frac{dM_{Pol}}{dt} = -F_{out}^{Pol} + r \quad (3.4)$$

O balanço de solvente no reator é dado por:

$$\frac{dM_{Sol}}{dt} = F_{in}^{Sol} - F_{out}^{Sol} \quad (3.5)$$

O balanço de catalisador no reator é dado por:

$$\frac{dM_{Cat}}{dt} = F_{in}^{Cat} - F_{out}^{Cat} \quad (3.6)$$

A taxa de reação r é dada pela equação:

$$r = x_{Mon} x_{Cat} k \exp\left(\frac{-E_a}{R T}\right) \quad (3.7)$$

Em que a constante cinética aparente k é dada por:

$$k = \frac{k_0 M_t^2}{V} \quad (3.8)$$

A quantidade total de matéria no reator M_t é dada por:

$$M_t = M_{Mon} + M_{Pol} + M_{Sol} + M_{Cat} \quad (3.9)$$

A vazão total de alimentação do reator é dada por:

$$F_{in} = F_{in}^{Mon} + F_{in}^{Sol} + F_{in}^{Cat} \quad (3.10)$$

A vazão de saída do reator é equivalente a vazão total de entrada, conforme a equação:

$$F_{out} = F_{in} \quad (3.11)$$

As vazões de saída por componente são dadas por:

$$F_{out}^i = F_{out} x_i, \quad i = \{Mon, Pol, Sol, Cat\} \quad (3.12)$$

Em que a fração molar x é dada por:

$$x_i = \frac{M_i}{M_t}, \quad i = \{Mon, Pol, Sol, Cat\} \quad (3.13)$$

Os parâmetros do modelo (Tabela 3.1) foram ajustados manualmente, de modo a representar uma dinâmica típica de um reator do tipo tanque. Este modelo matemático foi utilizado como planta virtual e como modelo do controlador durante as simulações. O esquema básico de controle consiste em controlar a temperatura do reator manipulando a temperatura da camisa (detalhes na Seção 3.3).

Tabela 3.1: Parâmetros do modelo

Parâmetros	Valores	Unidades
k	$5,00 \times 10^7$	mol/h
E_a	10046,40	J/mol
R	8,31	J/mol.K
T_{in}	350,00	K
ΔH_r	25116,00	J/mol
C_p	125,58	J/mol.K
κ_c	1,00	1/h

A constante cinética k aparente da reação é função de M_t e V , conforme mostra a Equação 3.8. Como as vazões de entrada e saída do reator são iguais (Equação 3.11) e M_t é a soma dos valores das variáveis de estado do modelo (Equação 3.9), dada uma condição inicial, M_t será constante, pois a reação química é do tipo $A \rightarrow B$ e, portanto, não altera a quantidade de matéria do meio reacional. Considerando que não há variações de massa específica, o volume V também é constante. Por isso, o lado direito da Equação 3.8 foi especificado como uma única constante, definida na Tabela 3.1. A mesma lógica foi aplicada à constante de troca térmica κ_c , reduzindo a ordem de grandeza do termo e, conseqüentemente, facilitando sua estimação (detalhes sobre a estimação na Seção 3.4).

Neste modelo não foram analisados parâmetros de qualidade típicos de sistemas de polimerização (como distribuição de massas molares, distribuição de tamanho de cadeia e índice de polidispersão) porque o foco deste trabalho se concentra no algoritmo de segurança. O uso de modelos mais sofisticados, no entanto, é sugerido para trabalhos futuros.

3.2 Métodos numéricos

Este trabalho foi inteiramente desenvolvido no *software* MATLAB (MATLAB, 2011) e os códigos podem ser encontrados nos Apêndices. Todas as simulações dinâmicas foram realizadas com o integrador numérico DASSLC (SECCHI, 1992–2007). A função `fsolve` foi utilizada para resolver o sistema de equações algébricas não lineares, obtido a partir do modelo estacionário, aplicando o algoritmo de Levenberg-Marquardt com tolerância (`TolFun`) de 10^{-9} . Os problemas de programação não linear (NLP – *Nonlinear Programming*) com restrições foram resolvidos com o algoritmo do Ponto Interior, através da função `fmincon`. As tolerâncias da função objetivo, variáveis de otimização e restrições (`TolFun`, `TolX` e `TolCon`) são, respectivamente, 10^{-7} , 10^{-6} e 10^{-5} para o controlador, e 10^{-9} e 10^{-8} para o reconciliador. Como o problema de reconciliação não possui restrições de igualdade ou desigualdade, a última tolerância não se aplica.

Os limites considerados para as variáveis manipuladas, no controlador, e para o parâmetro κ_c , no reconciliador, são mostrados na Tabela 3.2.

Tabela 3.2: Limites das variáveis

Variáveis	Limite	
	Inferior	Superior
T_c	300	420
F_{in}^{Mon}	0	200
F_{in}^{Sol}	0	400
F_{in}^{Cat}	0	2
κ_c	0	2

3.3 Controle Preditivo Não Linear

O Controle Preditivo Não Linear (NMPC – *Nonlinear Model Predictive Control*) é uma estratégia de controle avançado baseado em modelo não linear. Este controlador surgiu a partir do conceito do *Model Predictive Control* (MPC), idealizado na década de 60, e consiste basicamente em um problema de otimização dinâmica com múltiplos objetivos e restrições (GARCIA *et al.*, 1989). O NMPC vem ganhando destaque na indústria nos últimos anos, especialmente, por causa de três características (MANENTI, 2011): (i) é capaz de lidar com não linearidades econômicas e na dinâmica do processo; (ii) pode se basear em modelos matemáticos fenomenológicos ou modelo não lineares semi-empíricos; (iii) é capaz de resolver, simultaneamente, o problema de controle e o problema de otimização dinâmica em tempo real (problema

econômico), satisfazendo restrições para as variáveis de estado e para as variáveis manipuladas.

O NMPC encontra resistência de parte da indústria em virtude das dificuldades de implementação. Além das dificuldades de modelagem, há também a complexidade do algoritmo em si. Contudo, os benefícios da aplicação do controlador em processos com destacadas não linearidades são bem conhecidos (VIEIRA *et al.*, 2003; MANENTI, 2011; SOUZA *et al.*, 2011).

A rotina de otimização calcula M ações de controle para uma variável manipulada (Equação 3.14) e avalia a resposta do modelo por P tempos de amostragem (Equação 3.15). Como P é sempre maior que M , a última ação de controle calculada \mathbf{u}^{M-1} se repete até P . Usualmente, P é chamado de horizonte de predição e M é chamado de horizonte de controle.

$$\mathbf{u}^{calculado} = [\mathbf{u}^0, \mathbf{u}^1 \dots \mathbf{u}^{M-1}] \quad (3.14)$$

$$\mathbf{x}^{resposta} = [\mathbf{x}^1, \mathbf{x}^2 \dots \mathbf{x}^P] \quad (3.15)$$

A avaliação da resposta geralmente é feita em termos da diferença entre o valor da variável controlada e seu *setpoint* em cada ponto de amostragem, gerando um vetor de erros que, após ter cada elemento elevado ao quadrado, tem seus valores somados. O resultante é o valor da função objetivo do otimizador e, quanto menor for, mais próxima estará a variável controlada de seu *setpoint*. No ponto de mínimo tem-se o perfil ótimo em malha aberta para as variáveis manipuladas; contudo, devido a flutuações por ruído das variáveis medidas, distúrbios e incertezas no modelo, somente a primeira ação de controle calculada é aplicada na planta. Após isso o processo se repete. A nova estimativa inicial do otimizador é o vetor $\mathbf{u}^{calculado}$ deslocado, isto é, $\mathbf{u}^{guess} = [\mathbf{u}^1, \mathbf{u}^2 \dots \mathbf{u}^{M-1}, \mathbf{u}^{M-1}]$. O modelo do controlador é realimentado a cada instante de amostragem com os parâmetros estimados pelo reconciliador, que também fornece as condições iniciais para integração do modelo, método semelhante ao utilizado por SOUZA *et al.* (2011). A Figura 3.3 ilustra o funcionamento do MPC.

Na Figura 3.3, os pontos pretos representam as amostragens da variável de estado controlada do sistema e a linha sólida preta representa as ações de controle aplicadas no sistema. Suponha que o instante k é o estado atual do sistema e que uma mudança no *setpoint* da variável controlada é aplicada (linha tracejada). O MPC calcula M ações de controle, repetindo a última ação até P (linha sólida cinza). O modelo matemático é integrado e a resposta é avaliada em cada ponto de amostragem (pontos cinzas). Calculam-se, então, as diferenças quadráticas entre a resposta e o *setpoint* em cada ponto de amostragem e estes valores são somados gerando um

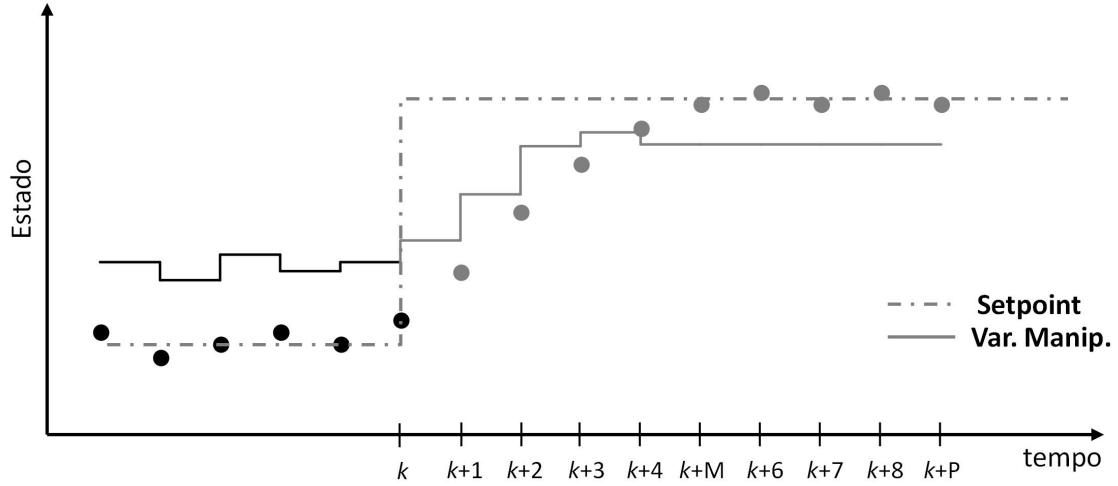


Figura 3.3: Ilustração do funcionamento do MPC.

escalar. Quanto menor este escalar, mais próxima estará a variável controlada de sua trajetória desejada. No ponto ótimo, a ação de controle calculada entre os instantes k e $k + 1$ é aplicada na planta e o processo se repete.

De maneira mais ampla, a formulação do controle preditivo também considera a supressão nas variáveis manipuladas e a perseguição de um ponto ótimo de operação determinada por uma camada de otimização. Assim, o problema de programação não linear (NLP) pode ser definido matematicamente da seguinte maneira (MEADOWS & RAWLINGS, 1997):

$$\begin{aligned} \min_{\mathbf{u}} J(k) = & \sum_{i=1}^P (\mathbf{y}(k+i|k) - \mathbf{y}_{sp}(k+i|k))^T \mathcal{Q} (\mathbf{y}(k+i|k) - \mathbf{y}_{sp}(k+i|k)) + \\ & \sum_{i=0}^M (\mathbf{u}(k+i|k) - \mathbf{u}_{sp}(k+i|k))^T \mathcal{S} (\mathbf{u}(k+i|k) - \mathbf{u}_{sp}(k+i|k)) + \\ & \sum_{i=0}^M \Delta \mathbf{u}(k+i|k)^T \mathcal{R} \Delta \mathbf{u}(k+i|k) \end{aligned} \quad (3.16)$$

sujeito às restrições: ($i = k, \dots, k + P - 1$)

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i) \quad (3.17)$$

$$\mathbf{y}_{i+1} = h(\mathbf{x}_{i+1}, \mathbf{u}_{i+1}) \quad (3.18)$$

$$\mathbf{u}_i \in \mathbb{U} \quad (3.19)$$

$$\mathbf{x}_i \in \mathbb{X} \quad (3.20)$$

$$\Delta \mathbf{u}_{max} \geq \Delta \mathbf{u} \geq \Delta \mathbf{u}_{min} \quad (3.21)$$

A função objetivo $J(k)$ (Equação 3.16) deve ser minimizada por meio da manipulação da variável de otimização \mathbf{u} , que é o vetor de ações de controle. Esta função objetivo é formada por três termos: o primeiro minimiza a diferença entre a variável controlada e seu *setpoint*; o segundo minimiza a diferença entre as ações de controle e suas trajetórias de referência (*targets*); e o terceiro minimiza a diferença entre as ações de controle calculadas no instante k e $k + 1$, $k + 1$ e $k + 2$ e etc, produzindo um efeito mais suave no perfil das ações de controle (termo de supressão de movimento). Existe também uma restrição rígida para $\Delta\mathbf{u}$, que garante uma variação máxima de ± 5 K entre cada instante de amostragem. Esta restrição foi implementada com o objetivo de aumentar a realidade das simulações, uma vez que existem limites práticos quanto ao controle regulatório da temperatura da camisa. Sem a restrição, variações de até 120 K (ver Tabela 3.2) seriam possíveis em apenas 3 min, que é o tempo de amostragem. Neste trabalho, as saídas \mathbf{y} são os próprios estados \mathbf{x} .

O controlador foi programado para assumir configurações distintas, dependendo da avaliação do risco ao qual o processo está sujeito (Tabela 3.3). Em condições normais de operação (operação segura) o controlador assume a configuração SISO (*Single Input Single Output*), em que a variável manipulada é a temperatura da camisa (T_c) e a variável controlada é a temperatura do reator (T). Quando o algoritmo de segurança detecta uma situação adversa (operação insegura), novos *setpoints* são calculados e, para atingi-los, o controlador é reconfigurado como MIMO (*Multiple Input Multiple Output*). Neste caso, as variáveis manipuladas são: temperatura da camisa (T_c), vazão de monômero (F_{in}^{Mon}), vazão de catalisador (F_{in}^{Cat}) e vazão de solvente (F_{in}^{Sol}) (não foram impostas restrições rígidas para as vazões). Já as variáveis controladas são a temperatura do reator (T), a massa de monômero no reator (M_{Mon}), a massa de catalisador no reator (M_{Cat}) e a massa de solvente no reator (M_{Sol}). Devido à reação química, monômero e polímero não podem ser controlados simultaneamente. Apenas um deles pode ser controlado, enquanto o outro tem seu comportamento vinculado à reação química.

Tabela 3.3: Configuração assumida pelo controlador em função do risco de operação

Operação segura - SISO		Operação insegura - MIMO	
VM	VC	VM	VC
T_c	T	T_c	T
		F_{in}^{Mon}	M_{Mon}
		F_{in}^{Cat}	M_{Cat}
		F_{in}^{Sol}	M_{Sol}

Vale ressaltar que o controlador MIMO é mantido até que o *setpoint* de segurança seja atingido. Após a convergência, a configuração retorna para a configuração

SISO, pois o algoritmo garante que esta configuração é capaz de manter o sistema em segurança.

3.4 Estimação de estados e parâmetros em tempo real

Para que o modelo matemático represente adequadamente o comportamento do reator, o modelo deve ser alimentado com informações que caracterizem o sistema, como concentrações de reagentes e produtos, temperatura, pressão, coeficiente global de troca térmica, constante de velocidade de reação, etc. Como o modelo é a base do controlador, as informações devem ser passadas a cada instante de amostragem de maneira precisa; caso contrário, o modelo seria incapaz de prever o comportamento do equipamento.

Essas informações são obtidas por meio de medições físicas, sejam diretas ou indiretas, e, portanto, são naturalmente corrompidas por erros de medição (PRATA *et al.*, 2009). Soma-se a isso ainda o ruído causado por perturbações não controladas das variáveis operacionais. Conseqüentemente, grande parte dos dados obtidos na planta não podem ser usados diretamente para alimentar o modelo, sob pena de não convergência ou obtenção de resultados imprecisos. Isto ocorre porque os balanços de massa e energia não são obedecidos por dados ruidosos. Para minimizar este problema, os dados obtidos devem ser reconciliados antes de alimentar o modelo. A reconciliação de dados é um procedimento de ajuste dos valores medidos, de forma que o conjunto reconciliado obedeça às leis de conservação e demais restrições físicas do sistema (PRATA *et al.*, 2009). Outro problema é que a integração numérica do modelo requer o conhecimento de todos os estados; porém, nem todos podem ser medidos. Portanto, os estados \mathbf{x} , assim como os parâmetros não medidos θ , precisam ser estimados para que o Problema de Valor Inicial (PVI) tenha solução aderente à planta (PRATA *et al.*, 2009).

De uma maneira geral, o sistema pode ser representado pelo diagrama de blocos da Figura 3.4. As variáveis medidas são amostradas periodicamente e enviadas ao reconciliador, que estima os estados e parâmetros do modelo do controlador. Com o modelo atualizado, o controlador calcula o próximo valor da variável manipulada, que é aplicado na planta e o ciclo se repete.

Em grande parte dos processos químicos ocorrem mudanças nos parâmetros ao longo da operação. Nesses casos, o procedimento de reconciliação utiliza uma janela móvel, também chamado horizonte móvel. Esta técnica foi apresentada por JANG *et al.* (1986) e representa uma redução do tamanho do problema, uma vez que divide o horizonte total em horizontes menores, como mostrado na Figura 3.5. Esta divisão

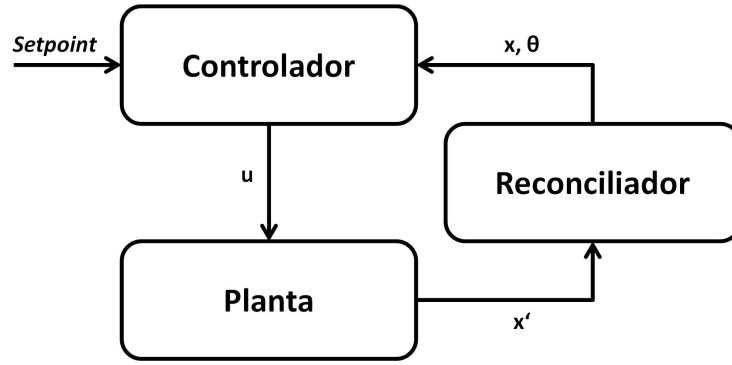


Figura 3.4: Diagrama de blocos do sistema Planta+Reconciliador+Controlador.

favorece a identificação do comportamento mais recente do processo, permitindo que o modelo se adapte continuamente às mudanças do processo. O tamanho da janela é um parâmetro de ajuste. Janelas pequenas podem levar a comportamentos oscilatórios (LIEBMAN *et al.*, 1992) e janelas longas elevam o custo computacional e podem até inviabilizar a aplicação em tempo real.

Uma janela é composta por um número H de pontos. Considerando k o instante atual, os pontos utilizados na reconciliação serão $k, k-1, k-2, \dots, k-H+1$. Portanto, quando a janela se move, o ponto mais antigo é descartado e o novo ponto é adicionado. A estimativa inicial da variável de otimização também é atualizada, sendo substituída pelo valor estimado mais recente.

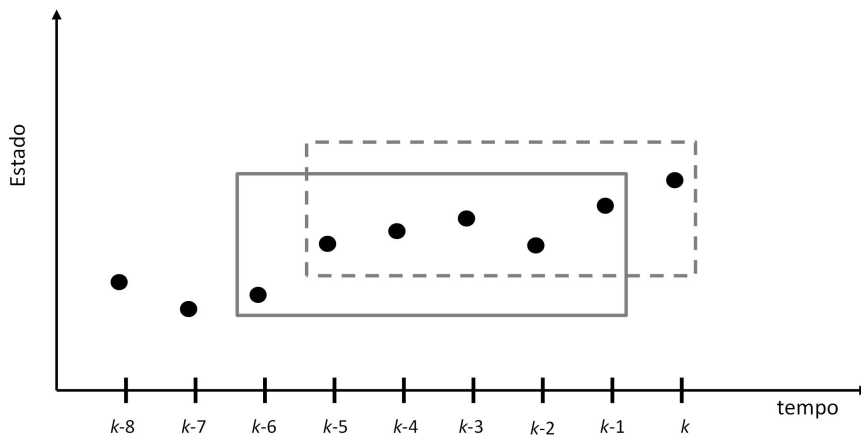


Figura 3.5: Horizonte móvel.

O problema de reconciliação dinâmica e não linear pode ser representado matematicamente da seguinte maneira (PRATA *et al.*, 2009):

$$\min_{\theta} \Phi(k) = \sum_{i=H_{rec}-1}^0 (\mathbf{y}(k-i|k) - \hat{\mathbf{y}}(k-i|k))^T (\mathbf{y}(k-i|k) - \hat{\mathbf{y}}(k-i|k)) \quad (3.22)$$

Sujeito às restrições: $(l = k - H_{rec} + 1, \dots, k - 1)$

$$\mathbf{x}_{l+1} = f(\mathbf{x}_l, \mathbf{u}_l, \theta) \quad (3.23)$$

$$\mathbf{y}_{l+1} = h(\mathbf{x}_{l+1}, \mathbf{u}_{l+1}, \theta) \quad (3.24)$$

$$\theta_k \in \mathbb{O} \quad (3.25)$$

Variando os parâmetros do vetor θ e simulando a resposta do modelo dinâmico de $k - H_{rec} + 1$ até k , a soma das diferenças ao quadrado entre as trajetórias estimadas e medidas em cada ponto de amostragem pode ser minimizada. Esta técnica é chamada de abordagem sequencial (ALICI & EDGAR, 2002). Cada parâmetro do vetor θ possui limites definidos. No ponto de mínimo (ponto ótimo) tem-se o valor estimado de θ e das variáveis de saída \mathbf{y} , que serão enviadas ao controlador para atualização do modelo e das condições iniciais da integração.

A temperatura da camisa e as vazões de alimentação e saída do reator são consideradas medidas perfeitas; isto é, sem ruído. Apenas a temperatura do reator foi adicionado um ruído gaussiano, com o objetivo de simular uma medida real no ambiente industrial. Portanto, neste trabalho, a função objetivo do reconciliador considera apenas a temperatura do reator “medida” e a temperatura do reator predita pelo modelo. Vale destacar que, neste trabalho, somente o parâmetro de transferência de calor com a camisa κ_c é estimado no procedimento de reconciliação e estimação simultâneos.

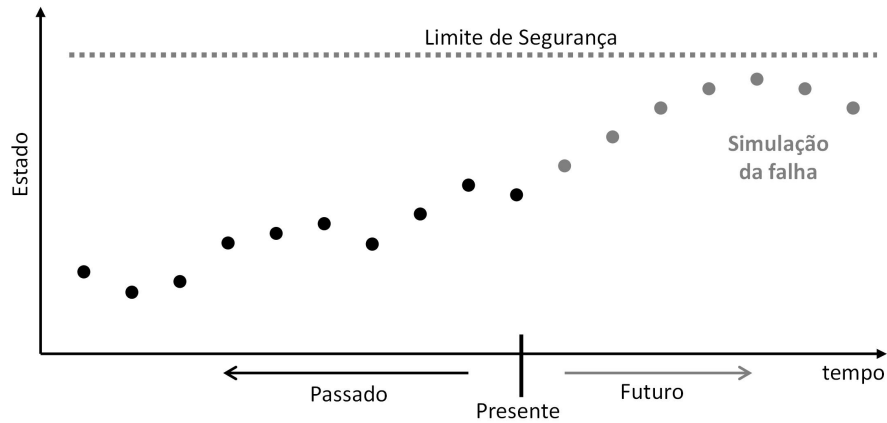
3.5 O sistema inteligente de segurança

O sistema inteligente de segurança tem por objetivo testar a capacidade real de reação do controlador frente a falhas críticas e tomar medidas de segurança antecipadas em caso de risco. Não há abordagens semelhantes na literatura, fato que torna a abordagem proposta inédita. Tem-se, portanto, duas etapas: uma para avaliar o risco e outra para determinar a medida de segurança.

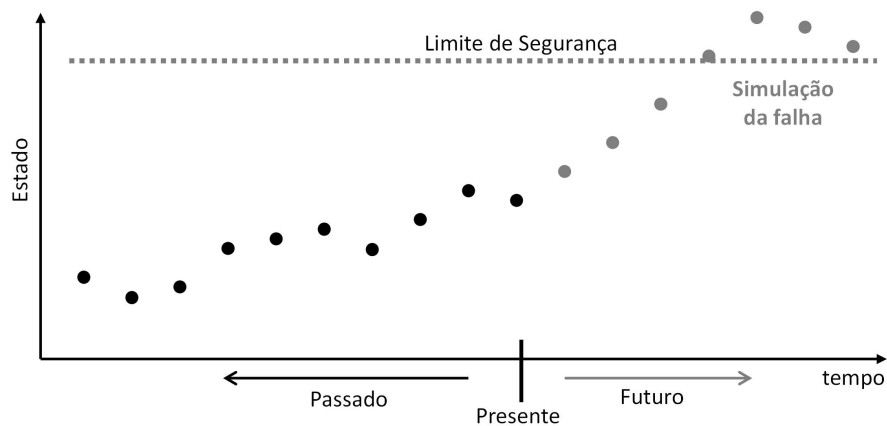
Etapa 1 – Avaliação do risco

Suponha que uma falha considerada comum ocorra no processo. Quais serão suas consequências? Diante dessas consequências, é seguro operar nesta região? Qual o limite a partir do qual haverá a perda do controle do processo? Para responder todas essas perguntas esta dissertação propõe o uso da simulação dinâmica. Com um modelo matemático confiável, é possível simular o comportamento da malha de controle do processo frente a uma falha. A trajetória calculada para as variáveis controladas é analisada e, se alguma delas violar os limites de segurança, a operação é considerada insegura. A Figura 3.6 ilustra este procedimento.

Etapa 2 – Medida preventiva de segurança



(a) Operação Segura



(b) Operação Insegura

Figura 3.6: Ilustração da simulação de falhas.

Após a constatação de que a operação é insegura, a primeira pergunta que se tem é: algo pode ser feito para tornar a operação segura e evitar um *shutdown* de emergência? Para responder esta pergunta, este trabalho propõe uma metodologia inédita capaz de dizer se existe, ou não, um estado seguro. A Figura 3.7 ilustra este procedimento.

A Figura 3.8 ilustra a integração do sistema inteligente de segurança ao sistema básico de controle e reconciliação. Na operação normal, as variáveis medidas são amostradas e enviadas ao reconciliador, que estima o parâmetro κ_c e todos os estados do modelo e atualiza o modelo do controlador, permitindo que o mesmo calcule as próximas ações de controle para perseguir o *setpoint*. Periodicamente, os dados do reconciliador são enviados ao simulador de falhas para análise de risco. Se não for detectado risco de operação, o processo segue normalmente. Se uma situação de risco potencial for detectada, o buscador de estados seguros será acionado. Se o buscador não encontrar um ponto de operação seguro, o *shutdown* de emergência torna-se a única alternativa para o caso de uma falha real na planta. Em contrapartida, se o

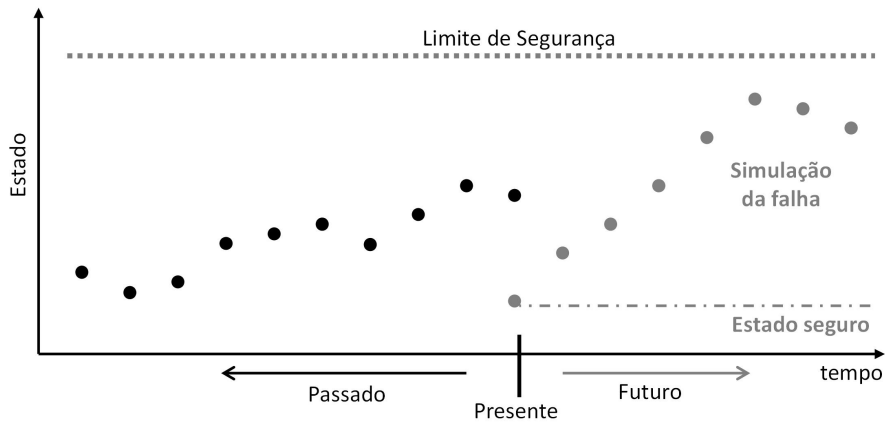


Figura 3.7: Ilustração do estado seguro.

buscador encontrar, o controlador pode ser reconfigurado para perseguir os novos *setpoints*, tornando o sistema seguro mesmo para o caso de uma falha real na planta. Em outras palavras, a decisão de manter a operação ou não, é responsabilidade da engenharia de processo. O sistema de segurança proposto traz uma alternativa de operação, que pode ou não ser seguida. Caso opte-se por não segui-la, corre-se o risco de haver um *shutdown* de emergência, que causaria grandes perdas econômicas.

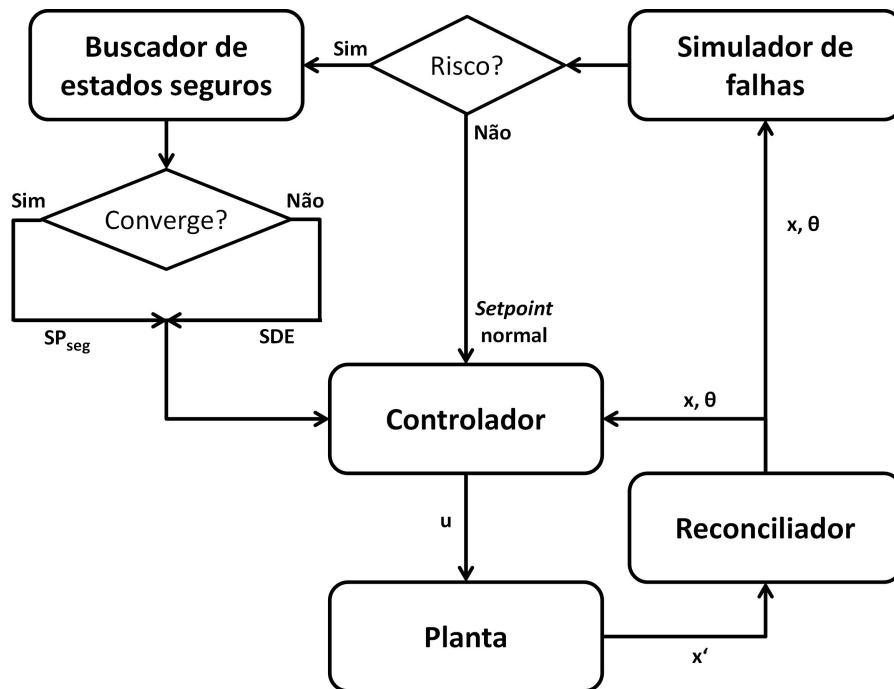


Figura 3.8: O sistema integrado de segurança.
 SDE: *Shutdown* de Emergência. SP_{seg} : *Setpoint* Seguro.

3.5.1 O simulador de falhas

O objetivo desta rotina é prever o comportamento do controlador diante de falhas e mapear os incidentes que podem causar perda de controle. O simulador de falhas é uma rotina capaz de simular diversas falhas a partir do estado atual da planta. Como se trata de um problema de valor inicial (PVI), é necessário conhecer todos os estados do modelo. Contudo, não é possível medir todos os estados e, mesmo que fosse possível, ainda haveria a contaminação por ruído. Sendo assim, é necessária a utilização de um reconciliador de dados. Em suma, o simulador de falhas funciona de maneira semelhante à própria planta e, portanto, pode ser representado pela Figura 3.4. As simulações podem ser executadas periodicamente, com período predefinido, ou quando for diagnosticada uma possível falha iminente.

O tempo de simulação considerado na rotina de simulação das falhas é um parâmetro de ajuste dependente do tempo característico de resposta do reator e do sistema de controle. Se o intervalo de tempo for pequeno, a resposta dinâmica será incompleta e a avaliação da severidade da falha ficará prejudicada. Um intervalo longo elevará o custo computacional, tornando a avaliação de risco inviável.

As trajetórias dinâmicas resultantes da simulação são comparadas com os limites de segurança do processo e, caso algum limite seja violado, o estado atual da planta é considerado um **estado de risco potencial**. Neste caso, um algoritmo para o cálculo da medida preventiva de segurança é acionado.

3.5.2 O buscador de estados seguros

O objetivo do buscador de estados seguros é encontrar um estado que, mesmo em caso de falha, seja possível manter o reator dentro dos limites de segurança. Este estado seguro deve ser o mais próximo possível do estado atual do reator, para que não haja um grande impacto na qualidade do produto.

Este algoritmo pode ser descrito como um problema de otimização dinâmica, em que as variáveis de otimização são as vazões de alimentação do reator (F_{in}^{Mon} , F_{in}^{Sol} e F_{in}^{Cat}) e a temperatura da camisa (T_c), ou seja, o próprio vetor \mathbf{u} . A função objetivo é a soma dos quadrados das diferenças entre o chamado *estado seguro* (vetor $\tilde{\mathbf{x}}(k + H_{cv})$) e o estado atual da planta (vetor $\mathbf{x}_0(k)$).

O estado seguro possui duas características fundamentais: Uma delas se refere à controlabilidade; ou seja, deve-se garantir que o controlador conseguirá manter as variáveis controladas no estado seguro. Sem esta garantia, o controlador pode levar o sistema para pontos operacionais desconhecidos, gerando uma situação de risco. A outra característica é que o estado seguro deve ser um ponto operacional de segurança inerente, isto é, deve garantir que os limites de segurança não serão violados em caso de falha. Por isso, o problema se divide em duas etapas: a primeira

verifica a viabilidade da solução e a segunda verifica se a solução garante a segurança do sistema.

Primeira etapa: Dado o estado atual da planta $\mathbf{x}_0(k)$ e seu respectivo vetor \mathbf{u} , o otimizador encontra um vetor \mathbf{u}^{ot} tal que a trajetória dinâmica provocada pela perturbação \mathbf{u}^{ot} a partir da condição inicial $\mathbf{x}_0(k)$ converge para um estado estacionário $\tilde{\mathbf{x}}(k + H_{cv})$ (Equação 3.30).

Segunda etapa: $\tilde{\mathbf{x}}(k + H_{cv})$ é dito um estado seguro se a trajetória dinâmica provocada pela perturbação ω (falha), considerando \mathbf{u}^{ot} constante (exceto T_c , que decresce a uma taxa imposta de 5 K até saturar em 300 K), a partir da nova condição inicial $\tilde{\mathbf{x}}(k + H_{cv})$, não violar o limite de segurança \mathbf{x}^{max} e não for crescente no tempo final de simulação (Equação 3.31).

A necessidade de $\tilde{\mathbf{x}}(k + H_{cv})$ ser um estado estacionário reside no fato de serem diferentes as dinâmicas do sistema em malha aberta (considerada pelo buscador) e em malha fechada (operação), e é garantida pela Equação 3.33. Na segunda etapa, a redução progressiva da temperatura da camisa simula, em malha aberta, o comportamento do controlador em situação de risco. Uma vez que os comportamentos inseguros considerados neste trabalho levam ao aumento da temperatura do reator, tem-se por objetivo avaliar a capacidade máxima de refrigeração do sistema. O horizonte H_{cv} é o tempo máximo disponível para levar a planta do estado atual $\mathbf{x}_0(k)$ para o estado seguro $\tilde{\mathbf{x}}(k + H_{cv})$ em malha aberta. Como a dinâmica imposta pelo controlador em malha fechada é mais rápida que a dinâmica em malha aberta, a convergência dentro da janela de tempo H_{cv} é garantida. (Para outras discussões sobre a dinâmica do controlador, consulte a Seção 4.2). O horizonte H_{sg} é o horizonte de avaliação da resposta da planta à falha a partir do estado seguro e deve ser suficiente para caracterizar a trajetória dinâmica.

Como a solução do problema de otimização é alcançada no menor valor da função objetivo sob restrição, o estado seguro é, conseqüentemente, o mais próximo possível do estado atual da planta. Matematicamente, o problema de otimização pode ser descrito como:

$$\min_{\mathbf{u}} \Psi = (\mathbf{x}_0(k) - \tilde{\mathbf{x}}(k + H_{cv}))^T \mathcal{X} (\mathbf{x}_0(k) - \tilde{\mathbf{x}}(k + H_{cv})) \quad (3.26)$$

sujeito às restrições:

$$\mathbf{x}(i+1) = f(\mathbf{x}(i), \mathbf{u}(i)) \quad i = k, \dots, k + H_{cv} - 1 \quad (3.27)$$

$$\mathbf{x}(k) = \mathbf{x}_0(k) \quad (3.28)$$

$$\check{\mathbf{x}}(j+1) = f(\mathbf{x}(j), \mathbf{u}(j); \omega) \quad j = k + H_{cv}, \dots, k + H_{cv} + H_{sg} - 1 \quad (3.29)$$

$$\check{\mathbf{x}}(k + H_{cv}) = \mathbf{x}(k + H_{cv}) \quad (3.30)$$

$$\check{\mathbf{x}}(k + H_{cv} + H_{sg}) - \check{\mathbf{x}}(k + H_{cv} + H_{sg} - 1) < Tol_{sg} \quad (3.31)$$

$$\check{\mathbf{x}}(j+1) < \mathbf{x}^{max} \quad (3.32)$$

$$|\mathbf{x}(k + H_{cv}) - \mathbf{x}(k + H_{cv} - 1)| < Tol_{cv} \quad (3.33)$$

$$\mathbf{u} \in \mathbb{U} \quad (3.34)$$

A variável ω representa uma falha do processo. Pode ser qualquer alteração que provoque um comportamento indesejado no sistema; por exemplo, alterações nos parâmetros da planta, desvios sistemáticos no \mathbf{u} medido ou aplicado na planta, etc. Neste trabalho são estudados os comportamentos provocados por alterações na constante de troca térmica, nas vazões de alimentação do reator e na temperatura da camisa (mais detalhes na Seção 4.4).

O funcionamento desta rotina pode ser resumido da seguinte maneira:

1. Fornecer uma estimativa inicial para \mathbf{u}
2. Simulação dinâmica do modelo (Equação 3.27)
3. Verificar se o módulo das diferenças de \mathbf{x} no tempo final são menores que Tol_{cv} para garantir que é um estado estacionário (Equação 3.33). Se não, corrigir o valor de \mathbf{u} pelo método do Ponto Interior e voltar para 2. Se sim, continuar
4. $\check{\mathbf{x}}$ é o estado estacionário obtido da simulação (Equação 3.30)
5. Simulação dinâmica do modelo com falha (Equação 3.29)
6. Verificar se a trajetória obtida da simulação permanece abaixo do limite \mathbf{x}^{max} (Equação 3.32) e se as diferenças de $\check{\mathbf{x}}$ no tempo final são menores que Tol_{sg} para garantir uma trajetória não crescente (Equação 3.31). Se não, corrigir o valor de \mathbf{u} pelo método do Ponto Interior e voltar para 2. Se sim, continuar
7. Estado seguro $\check{\mathbf{x}}(k + H_{cv})$ encontrado. Fim do algoritmo.

3.6 Comentários finais

O sistema em estudo é um típico reator CSTR (*Continuous Stirred Tank Reactor*) em que ocorre uma reação química do tipo $A \rightarrow B$ na presença de um solvente e de um catalisador. O solvente é inerte, mas o catalisador influencia na taxa

de reação. O controlador NMPC pode assumir configurações distintas em função da avaliação de risco do sistema. A única variável medida é a temperatura do reator, enquanto os demais estados são estimados, assim como a constante de troca térmica da camisa. As estimações são realizadas através de uma técnica de janela móvel, permitindo capturar com maior acurácia o comportamento mais recente da planta e possibilitando a atualização do modelo do controlador a cada intervalo de amostragem. As simulações de falhas levam em consideração os erros de medida e a dinâmica do sistema em malha fechada, o que permite uma previsão confiável do comportamento da planta em caso de falhas. Em situações de risco, a planta é levada para um estado seguro, cuja convergência é garantida pelo algoritmo, se afastando o mínimo possível do estado inicial. Não foram encontradas, na literatura científica, estratégias semelhantes na abordagem de sistemas de segurança.

Capítulo 4

Resultados e Discussões

Para discutir o desempenho do sistema de segurança, deve-se garantir que tanto o controlador quanto o reconciliador estão funcionando corretamente. Mas inicialmente, deve-se garantir que o modelo matemático representa adequadamente o sistema físico proposto. Desta forma, como introdução deste capítulo, simulações e análises dinâmicas do modelo são discutidas na Seção 4.1. Em seguida, os desempenhos dos sistemas de controle e reconciliação são discutidos nas Seções 4.2 e 4.3, respectivamente. Na Seção 4.5, são apresentados os resultados de simulações de diversos cenários construídos para avaliar o comportamento do algoritmo de segurança.

4.1 Dinâmica do sistema

As simulações desta seção mostram o comportamento dinâmico do sistema em malha aberta para perturbações degrau nas variáveis de entrada. O estado estacionário de partida é mostrado na Tabela 4.1. Este ponto de operação foi selecionado arbitrariamente, de forma que o tempo de residência fosse próximo a 1 h; valor comum em reatores de polimerização.

Tabela 4.1: Estado estacionário inicial para avaliação da dinâmica do sistema

Variáveis de estado	Valores iniciais	Variáveis de entrada	Valores iniciais
T	380,0	T_c	361,0
M_{Mon}	36,2	F_{in}^{Mon}	112,7
M_{Pol}	77,2	F_{in}^{Sol}	198,6
M_{Sol}	200,0	F_{in}^{Cat}	0,10
M_{Cat}	0,10		

Todas as perturbações degrau foram executados no tempo de 1 h, modificando apenas a variável de entrada em questão e mantendo as demais inalteradas. Por isso,

apenas a variável de entrada modificada é mostrada nos gráficos. Os valores finais assumidos nas perturbações degrau são referentes aos limites considerados para as variáveis de entrada do sistema, destacados na Tabela 3.2.

O tempo de amostragem utilizado nas simulações foi de 3 min. De uma forma geral, intervalos pequenos causam aumento do custo computacional e intervalos grandes causam perda de informações acerca da dinâmica da variável. O valor utilizado foi considerado adequado sob ambos os pontos de vista e compatível com o utilizado no meio industrial.

4.1.1 Efeito da temperatura da camisa

As respostas dinâmicas referentes aos efeitos de altas e baixas temperaturas da camisa são mostrados nas Figuras 4.1 e 4.2. O aumento da temperatura da camisa provoca o aumento da temperatura do reator que, por sua vez, aumenta a taxa de reação e, conseqüentemente, aumenta a conversão do monômero em polímero. Em contrapartida, a redução da temperatura da camisa provoca a redução da temperatura do reator e, conseqüentemente, reduz a conversão do monômero. Portanto, o comportamento do modelo está de acordo com o esperado.

4.1.2 Efeito da vazão de monômero

As respostas dinâmicas referentes aos efeitos de altas e baixas vazões de monômero são mostrados nas Figuras 4.3 e 4.4. O aumento da vazão de monômero provoca o aumento permanente da quantidade do reagente no reator (Figura 4.3b). Como a vazão de saída do reator é sempre igual à vazão total de entrada, o aumento da vazão do monômero causa um aumento proporcional na vazão de saída. Em consequência, ocorre uma redução das quantidades de solvente e catalisador no reator (Figuras 4.3c e 4.3d), visto que suas alimentações permanecem inalteradas. A temperatura do reator exibe uma resposta inversa bastante suave (Figura 4.3a), provavelmente, devido ao efeito inicial do monômero como diluente de calor, uma vez que o reagente é alimentado a 350 K; ou seja, mais frio que a massa reacional, que está a 380 K. Logo em seguida, há um aumento da temperatura do reator, possivelmente, em virtude da sobreposição do calor liberado pela reação química. Comportamento semelhante é observado na resposta da quantidade de polímero no reator (Figura 4.3e). Inicialmente há um efeito de “arraste” do polímero, uma vez que o produto não é alimentado e a vazão de saída do reator sofre um aumento brusco. Contudo, o aumento da taxa de reação promove a formação do polímero, efeito que prevalece no meio reacional.

Os efeitos da redução da vazão de monômero são qualitativamente semelhantes aos verificados para o aumento da vazão do reagente, entretanto, ocorrem de maneira

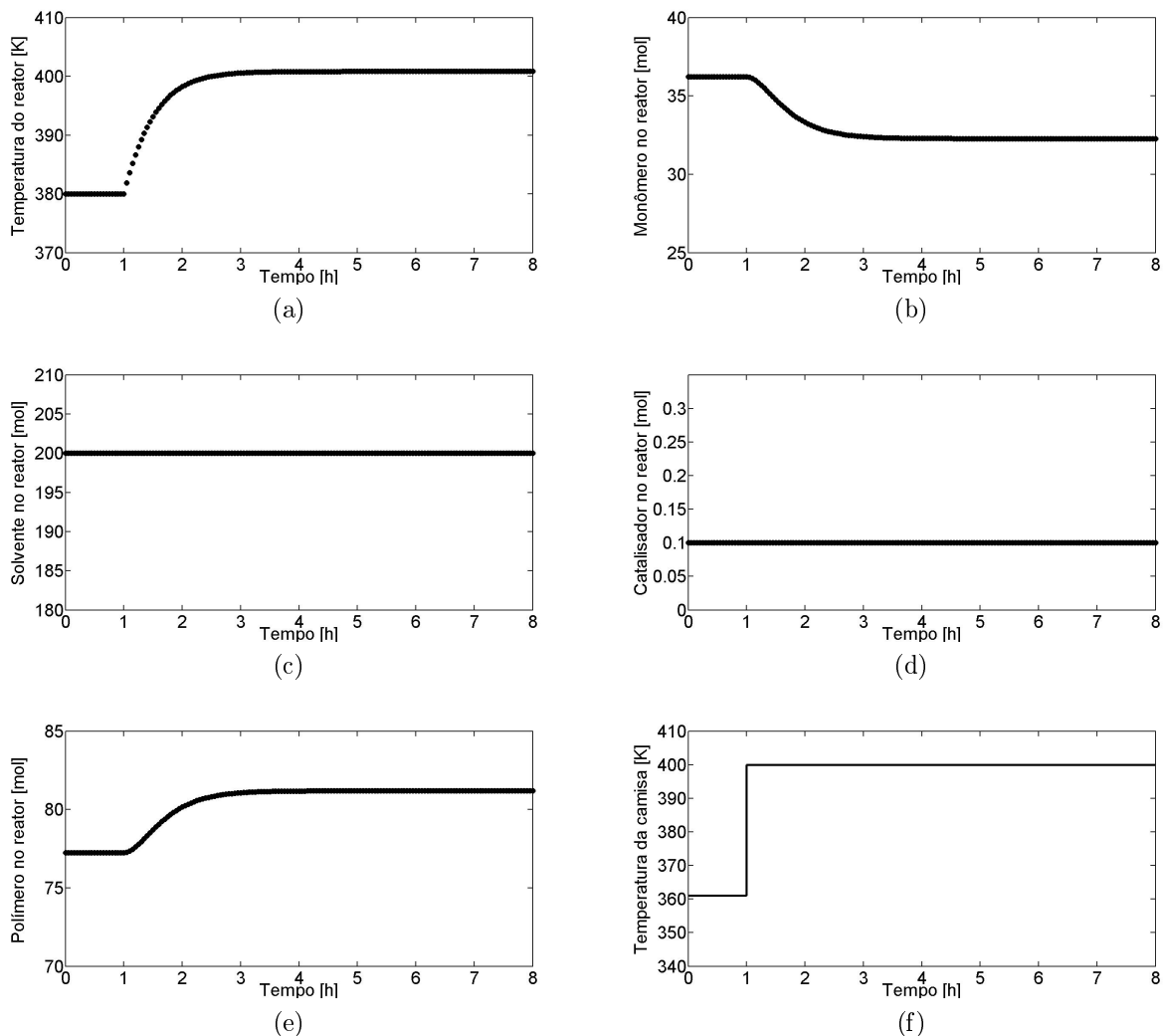


Figura 4.1: Dinâmica do modelo em malha aberta. Temperatura da camisa alta.

inversa. Com a redução da vazão de alimentação, há redução da vazão de saída, que provoca o acúmulo de solvente e catalisador no reator (Figuras 4.4c e 4.4d). O mesmo acontece inicialmente com o polímero (Figura 4.4e); contudo, a redução da quantidade de monômero no reator, causa redução na taxa de formação do polímero que, conseqüentemente, tem sua proporção reduzida no reator. Sem o calor gerado pela reação química, a temperatura do reator diminui (Figura 4.4a). Ambos os comportamentos são coerentes com o esperado.

4.1.3 Efeito da vazão de solvente

As respostas dinâmicas referentes aos efeitos de altas e baixas vazões de solvente são mostrados nas Figuras 4.5 e 4.6. O aumento da vazão de solvente provoca o aumento da vazão de saída do reator. Conseqüentemente, ocorre o arraste das substâncias que mantiveram suas alimentações inalteradas, por isso a redução das

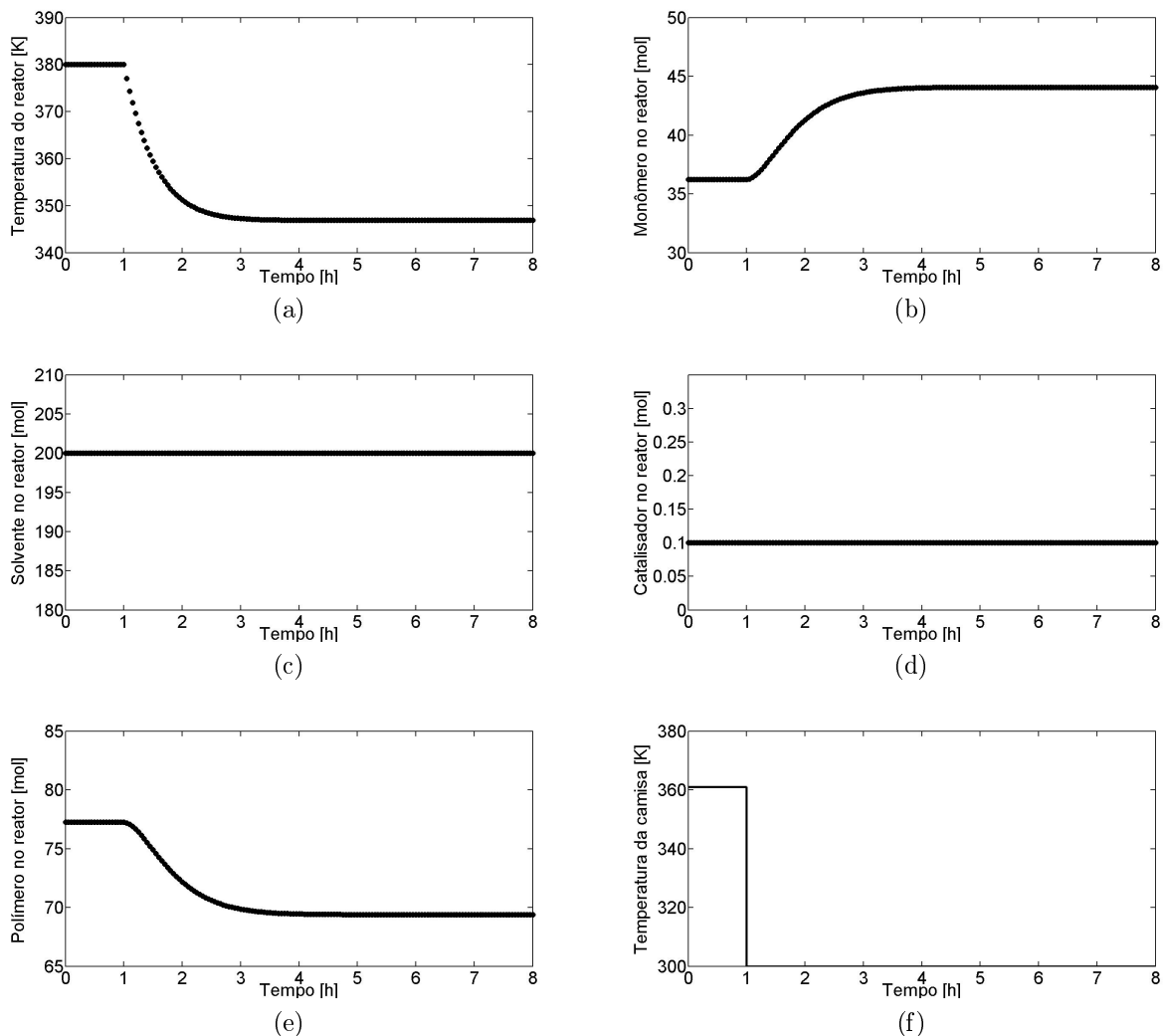


Figura 4.2: Dinâmica do modelo em malha aberta. Temperatura da camisa baixa.

quantidades de catalisador, de polímero e, inicialmente, de monômero (Figuras 4.5d, 4.5e e 4.5b). Em relação ao último, soma-se o efeito de diluição, que promove a redução da taxa de reação e, portanto, o acúmulo da substância no reator. Este efeito potencializa a redução da quantidade de polímero no meio reacional, proporcionando uma redução acentuada da substância. Com o aumento da corrente de alimentação, que é mais fria que o meio reacional, ocorre redução da temperatura do reator (Figura 4.5a). A redução da taxa de reação, por efeito da diluição, também colabora para o declínio da temperatura que, por sua vez, afeta negativamente a taxa de reação.

A redução da vazão de solvente provoca a redução da vazão de saída do reator. Isso provoca o acúmulo das substâncias cuja alimentação não foi alterada no meio reacional, como se pode notar para o catalisador e, inicialmente, para o monômero (Figuras 4.6d e 4.6b). Com a redução do solvente no reator e o aumento do catalisador, ocorre aumento da taxa de reação. Esta provoca o consumo do monômero e formação do polímero, com a conseqüente liberação de energia. Por isso há de-

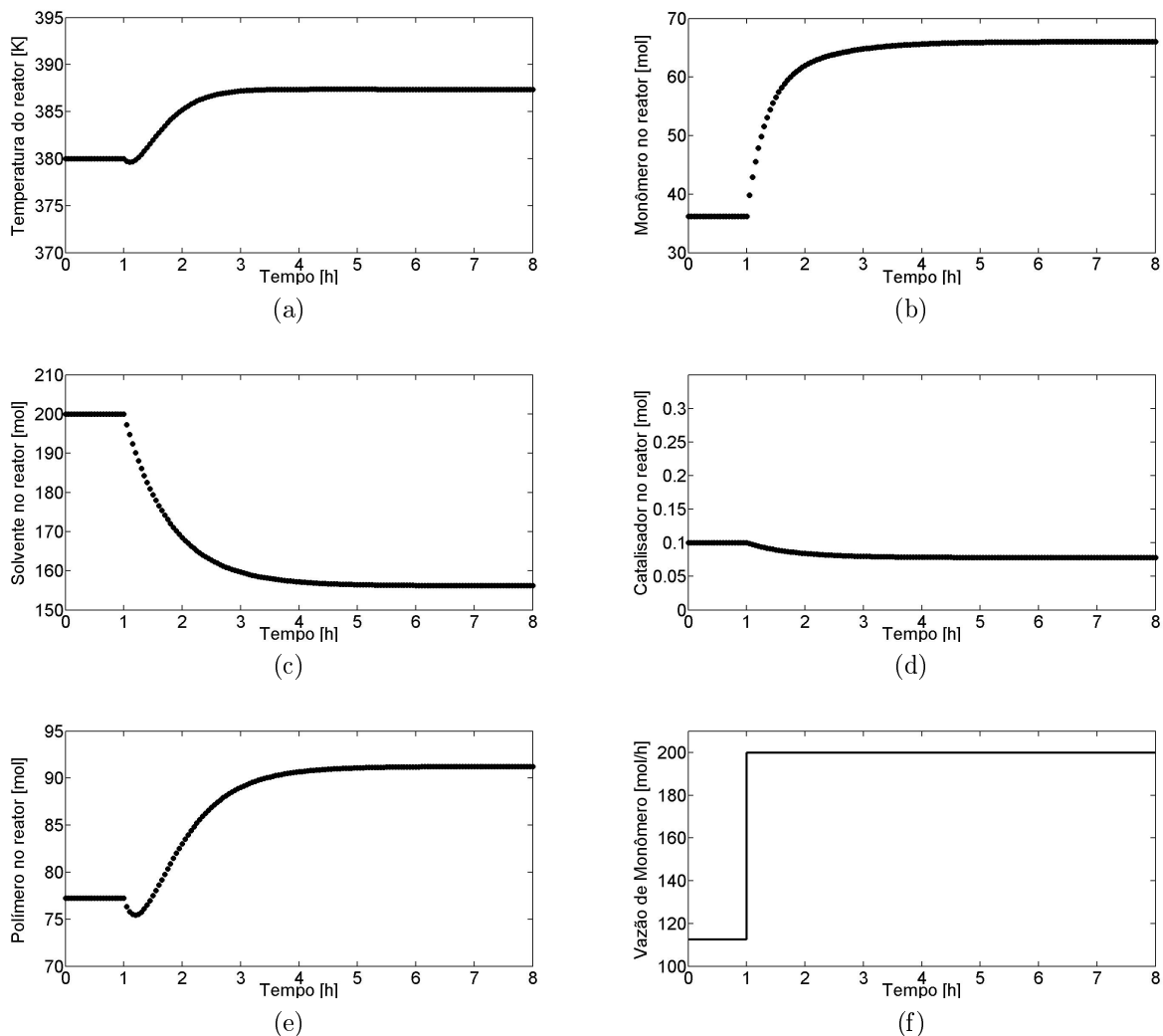


Figura 4.3: Dinâmica do modelo em malha aberta. Vazão de monômero alta.

créscimo permanente da quantidade de monômero e aumentos na quantidade de polímero e na temperatura do meio reacional (Figuras 4.6e e 4.6a).

4.1.4 Efeito da vazão de catalisador

As respostas dinâmicas referentes aos efeitos de altas e baixas vazões de catalisador são mostrados nas Figuras 4.7 e 4.8. O aumento da vazão de catalisador pouco influencia na vazão total de alimentação e, portanto, não tem grandes impactos na vazão de saída do reator. Isso faz com que a quantidade de solvente no reator permaneça quase inalterada (Figura 4.7c). Embora tenha pequena influência no tempo de residência do reator, o catalisador tem efeito destacado na taxa de reação. Tal influência pode ser observada pelo aumento da conversão da reação; isto é, diminuição da quantidade de monômero, aumento da quantidade de polímero e aumento da temperatura (Figuras 4.7b, 4.7e e 4.7a). Destaque para o *overshoot* na

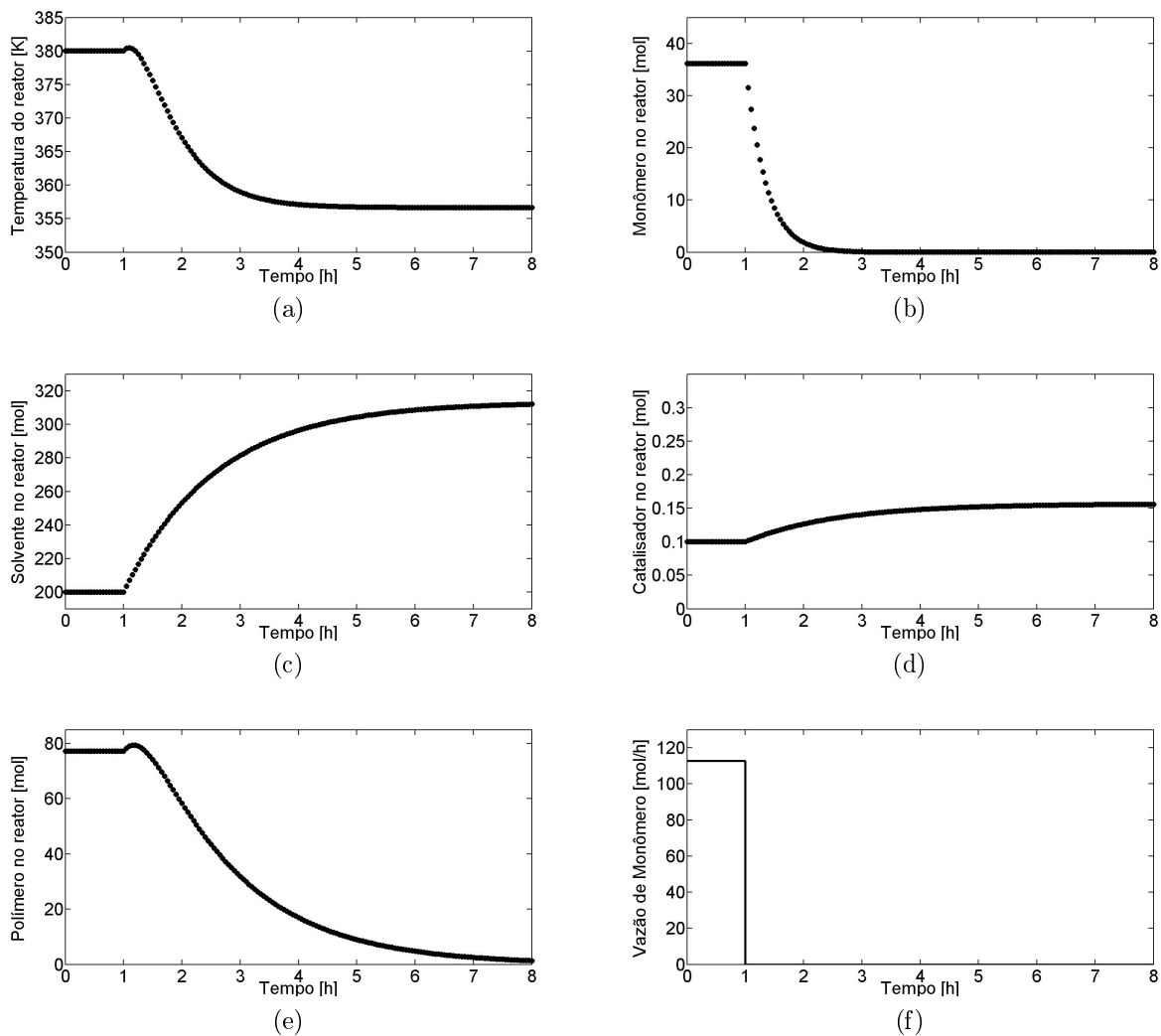
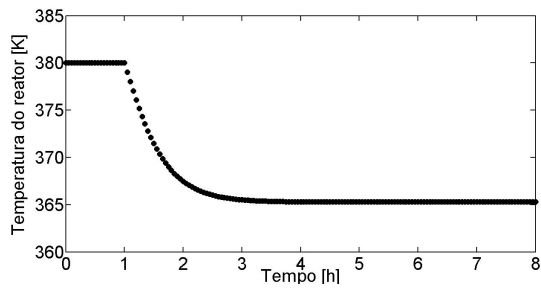


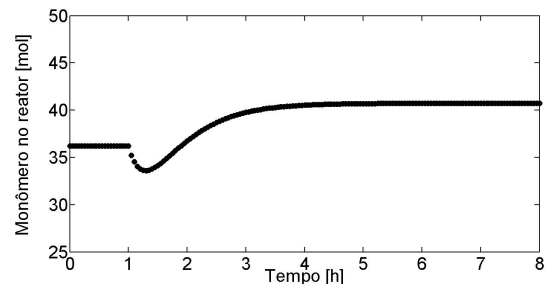
Figura 4.4: Dinâmica do modelo em malha aberta. Vazão de monômero baixa.

resposta da temperatura do reator.

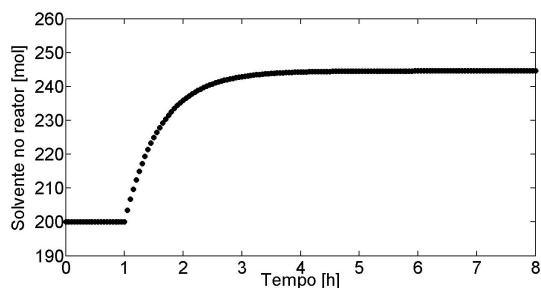
A redução da vazão de catalisador anula a taxa de reação, provocando acúmulo de monômero no reator (Figura 4.8b). Como não há reação, o polímero é arrastado do reator (Figura 4.8e) e a temperatura diminui (Figura 4.8a), em função da ausência de geração de energia. Todas as simulações produziram comportamentos dentro do previsto.



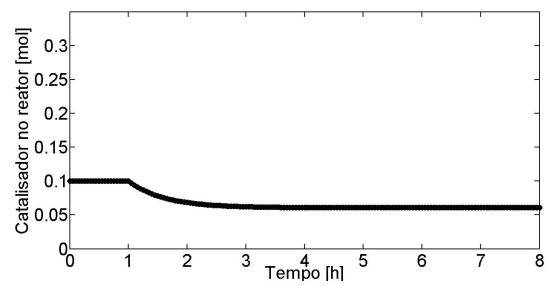
(a)



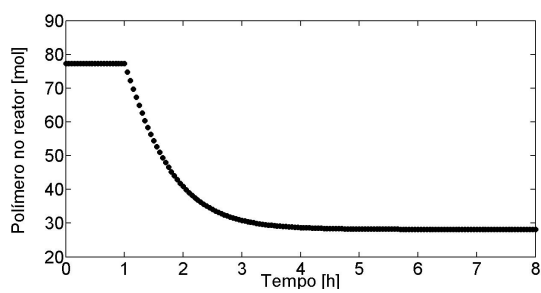
(b)



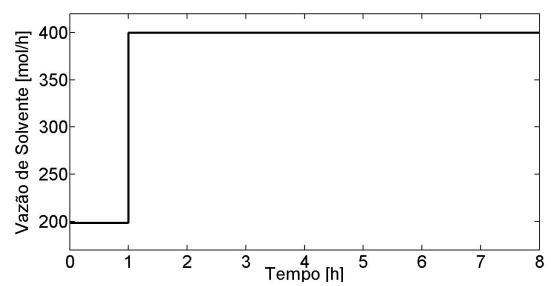
(c)



(d)

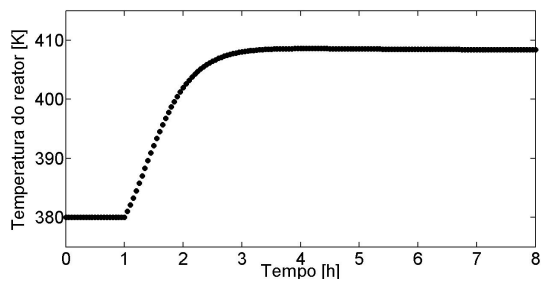


(e)

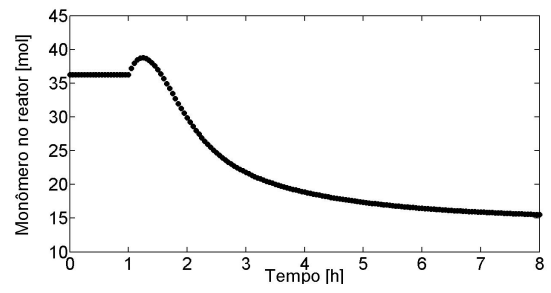


(f)

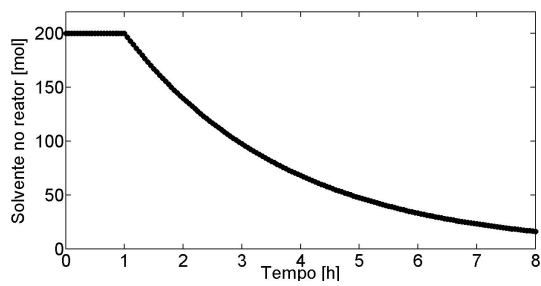
Figura 4.5: Dinâmica do modelo em malha aberta. Vazão de solvente alta.



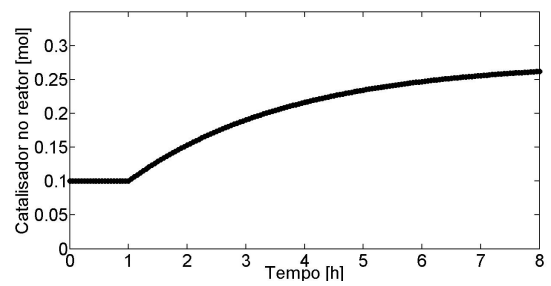
(a)



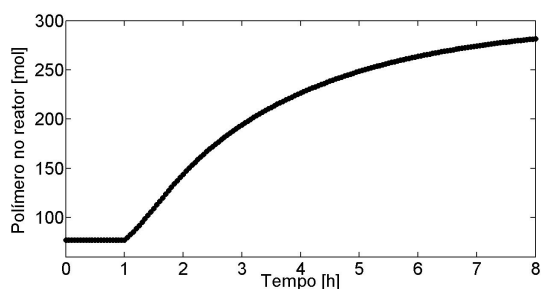
(b)



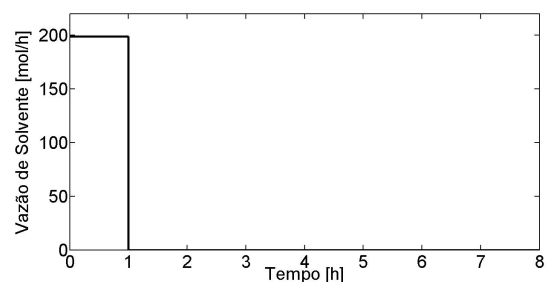
(c)



(d)

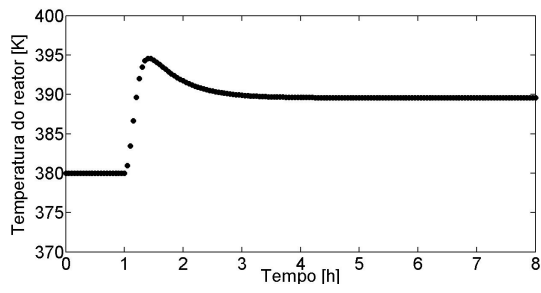


(e)

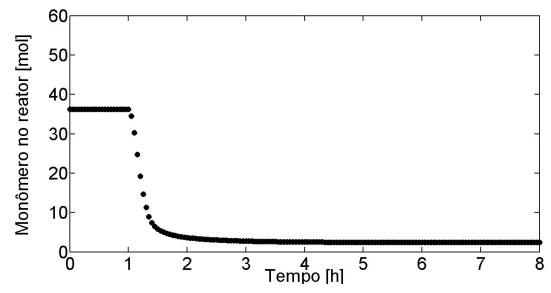


(f)

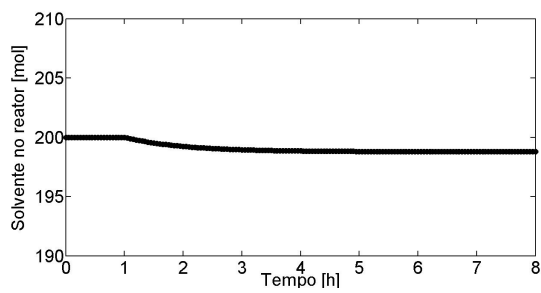
Figura 4.6: Dinâmica do modelo em malha aberta. Vazão de solvente baixa.



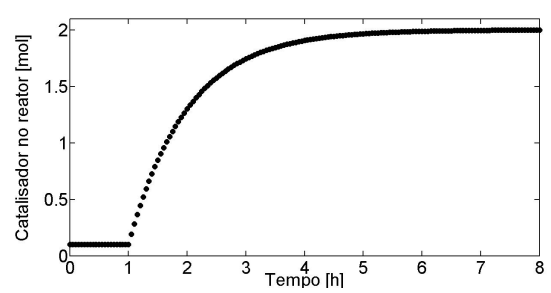
(a)



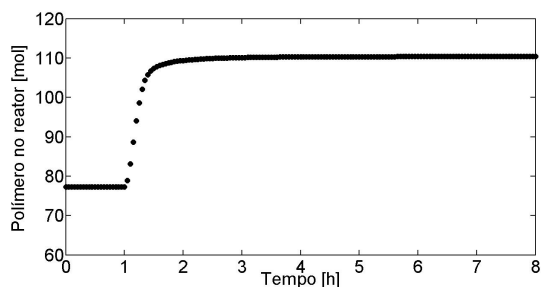
(b)



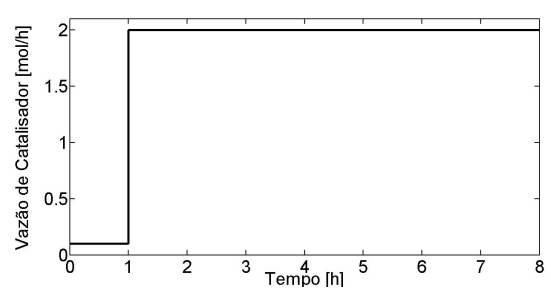
(c)



(d)

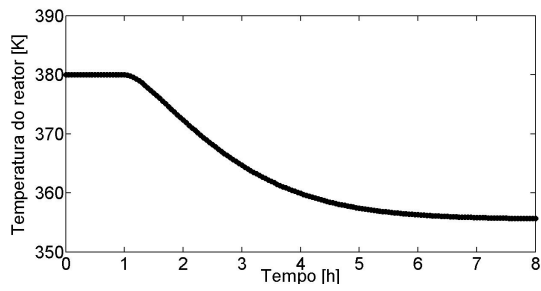


(e)

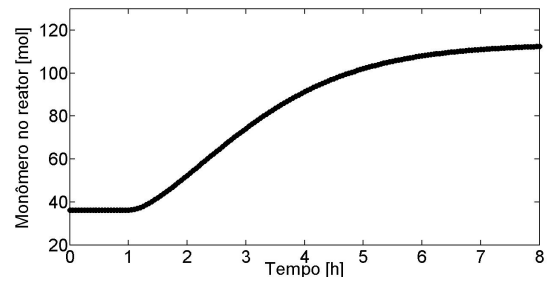


(f)

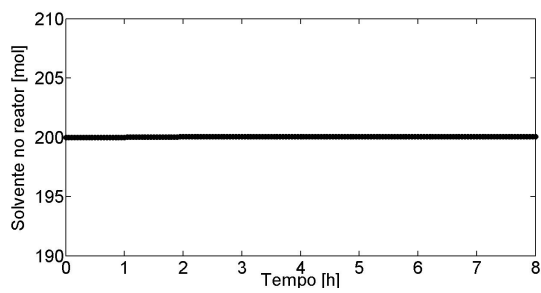
Figura 4.7: Dinâmica do modelo em malha aberta. Vazão de catalisador alta.



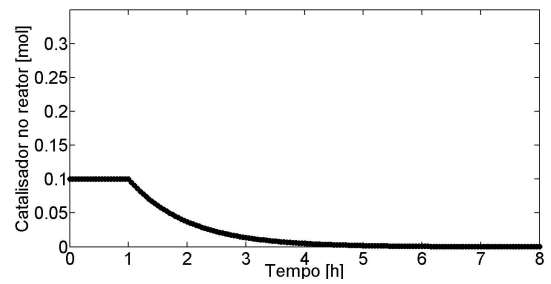
(a)



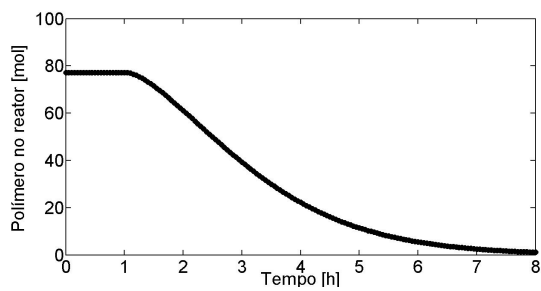
(b)



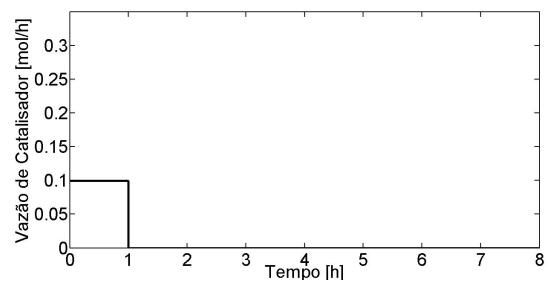
(c)



(d)



(e)



(f)

Figura 4.8: Dinâmica do modelo em malha aberta. Vazão de catalisador baixa.

4.2 Desempenho do controlador

Para o estudo do algoritmo de controle, dois casos distintos foram levados em consideração. Ambos os casos representam uma mudança no *setpoint* do controlador, em que todas as variáveis controladas devem ser levadas do Estado 1 para o Estado 2. Cada estado é constituído por um vetor \mathbf{x} e um vetor \mathbf{u} que são soluções do problema estacionário $\mathbf{x}(k) = f(\mathbf{x}(k), \mathbf{u}(k))$.

- ☞ *Caso 1: o Estado 2 é inatingível.* Significa que, a partir do Estado 1 $f(\mathbf{x}_1, \mathbf{u}_1)$, dada uma perturbação degrau \mathbf{u}_2 , o sistema dinâmico converge para o Estado 3 $f(\mathbf{x}_3, \mathbf{u}_2)$.
- ☞ *Caso 2: o Estado 2 é atingível.* Significa que, a partir do Estado 1 $f(\mathbf{x}_1, \mathbf{u}_1)$, dada uma perturbação degrau \mathbf{u}_2 , o sistema dinâmico converge para o Estado 2 $f(\mathbf{x}_2, \mathbf{u}_2)$.

Testar o desempenho do controlador nestes cenários é importante porque são situações que podem ocorrer durante a atuação do sistema de segurança, que, em situações de risco, deve deslocar o processo para um outro ponto de operação. Todavia, este novo estado pode se enquadrar no Caso 1, definido acima, gerando um *offset* em relação ao estado seguro calculado.

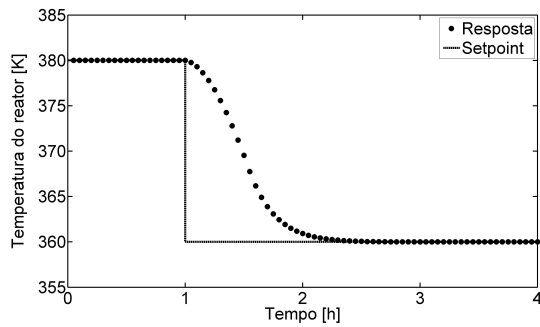
Partindo do estado estacionário definido pelo *Setpoint* 1 (Tabela 4.2), foram realizadas simulações dinâmicas visando a observar o comportamento do sistema frente a uma mudança no *setpoint* do controlador no instante 1 h para os valores definidos pelo *Setpoint* 2, que é inatingível (Caso 1). O desempenho do NMPC pode ser ajustado manipulando-se os parâmetros de sintonia; ou seja, horizonte de predição (P), horizonte de controle (M) e pesos associados aos termos da função objetivo (\mathcal{Q} , \mathcal{R} e \mathcal{S}).

O desempenho do controlador SISO para o caso *Setpoint* 1 \rightarrow *Setpoint* 2 é mostrado na Figura 4.9. Variações nos horizontes de controle e predição não proporcionaram melhores desempenhos, apenas aumento do custo computacional. Assim, optou-se por utilizar uma sintonia que proporcionasse um desempenho satisfatório e o menor custo computacional, neste caso, $[M, P] = [3, 10]$. Os respectivos pesos \mathcal{Q} e \mathcal{R} adotados são matrizes identidade. Embora não considerada no caso SISO, a influência do peso \mathcal{S} é discutida durante a sintonia do controlador MIMO. O desempenho do controlador neste caso foi considerado satisfatório, uma vez que não houve *offset* e o perfil da variável manipulada se mostrou suave. Pela simplicidade do controlador SISO, não foram realizados outros testes. Discussões mais detalhadas acerca da sintonia são tratadas adiante para a configuração MIMO.

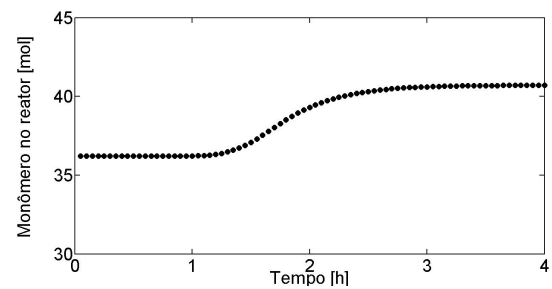
O desempenho do controlador MIMO é para o caso *Setpoint* 1 \rightarrow *Setpoint* 2 mostrado nas Figuras 4.10 e 4.11, sendo possível notar uma dificuldade moderada

Tabela 4.2: Estados da planta durante as simulações de desempenho do controlador.

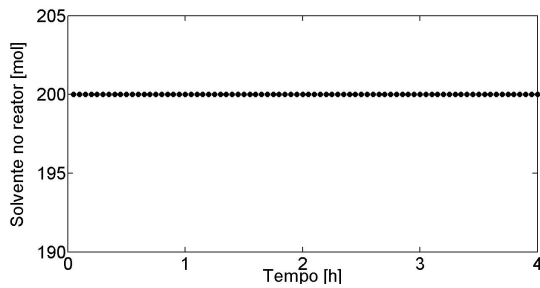
Variável	Setpoint 1	Setpoint 2 (Inatingível)	Setpoint 3 (Atingível)
T	380,0	360,0	362,7
M_{Mon}	36,2	25,0	22,2
M_{Pol}	77,2	114,7	104,6
M_{Sol}	200,0	210,0	190,6
M_{Cat}	0,10	0,30	0,27
T_c	361,0	308,6	308,6
F_{in}^{Mon}	112,7	130,1	130,1
F_{in}^{Sol}	198,6	195,4	195,4
F_{in}^{Cat}	0,10	0,28	0,28
κ_c	1,0	1,0	1,0



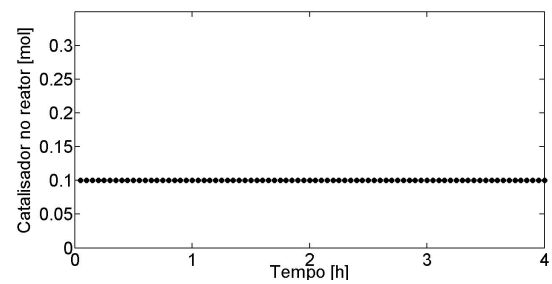
(a)



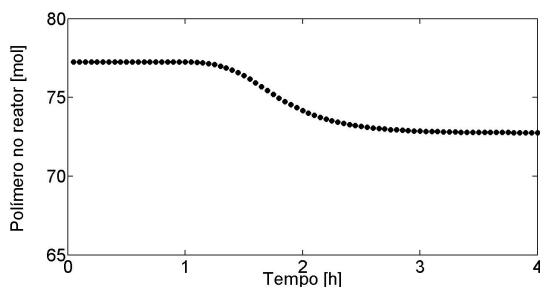
(b)



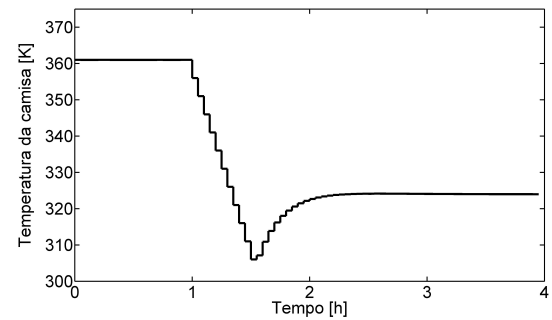
(c)



(d)



(e)



(f)

Figura 4.9: Desempenho do controlador SISO: Setpoint 1 \rightarrow Setpoint 2.

em controlar a quantidade de catalisador no reator (Figura 4.10d). Esta dificuldade é decorrente da cinética de reação, que é proporcional ao produto entre as frações de monômero e catalisador ($r = x_{Mon} x_{Cat} k$). Variações na fração de catalisador (x_{Cat}) provocam mudanças na taxa de reação (r) que, por sua vez, afeta a quantidade de monômero no reator (ver Equação 3.3). Já mudanças na fração de monômero, têm efeito apenas de mistura em relação ao catalisador, como se pode verificar pelo balanço global do reator ($M_t = M_{Mon} + M_{Pol} + M_{Cat} + M_{Sol}$) e pela ausência do termo de reação química no balanço por componente do catalisador (Equação 3.6).

A estratégia inicial para definição dos pesos \mathcal{Q} , \mathcal{R} e \mathcal{S} é ilustrada na Tabela 4.3. Os pesos foram definidos de modo que todas as variáveis exercessem impactos similares na função objetivo. Tendo como referência o peso atribuído à temperatura do reator T , os outros pesos foram calculados de modo que o produto entre os valores da coluna “Variação máxima” e da coluna “Peso” totalizasse 100. Assim, as variações de todas as variáveis exercem efeitos similares na função objetivo. Como a quantidade de polímero no reator não é controlada, a variável não afeta a função objetivo do controlador, por isso foi omitida na Tabela 4.3. Inicialmente, a função objetivo do controlador MIMO considera apenas o primeiro e último termos da Equação 3.16, assim como nas simulações anteriores do controlador SISO, portanto, a influência do peso \mathcal{S} não foi avaliada nesta sintonia, embora sua definição inicial tenha seguido a mesma lógica do peso \mathcal{R} .

Tabela 4.3: Estratégia de seleção dos pesos da função objetivo do controlador

Parâmetro	Var. Associada	Variação máxima	Peso
\mathcal{Q}	T	100	1
	M_{Mon}	100	1
	M_{Sol}	200	0,5
	M_{Cat}	1	100
\mathcal{R}	T_c	100	1
	F_{in}^{Mon}	200	0,5
\mathcal{S}	F_{in}^{Sol}	400	0,25
	F_{in}^{Cat}	1	100

Nota-se na Figura 4.10c que ocorreu um leve *overshoot* no desempenho do solvente. A fim de tentar reduzir a amplitude de tal desvio, o peso referente ao solvente foi dobrado, passando, portanto, de 0,5 para 1. Desta forma, aumenta-se a influência do erro de *setpoint* do solvente na função objetivo do controlador, resultando em uma grande diminuição do *overshoot*, como mostrado nas Figuras 4.12 e 4.13.

Em ambas as sintonias observa-se um destacado *offset* no *setpoint* do catalisador, cuja trajetória dinâmica começa com um pequeno afastamento do valor desejado, mas em seguida retoma seu comportamento esperado. O *offset* ocorre porque o

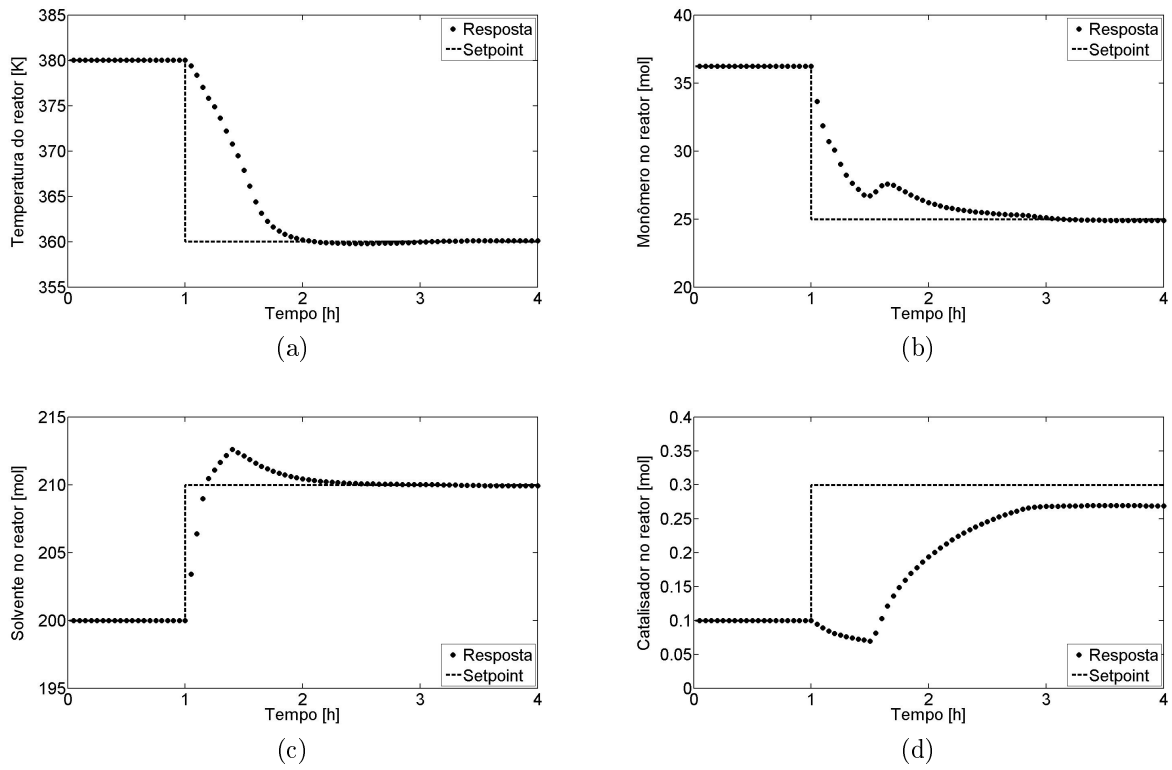


Figura 4.10: Desempenho do controlador MIMO: $Setpoint\ 1 \rightarrow Setpoint\ 2$. Primeira sintonia do peso Q . Variáveis controladas.

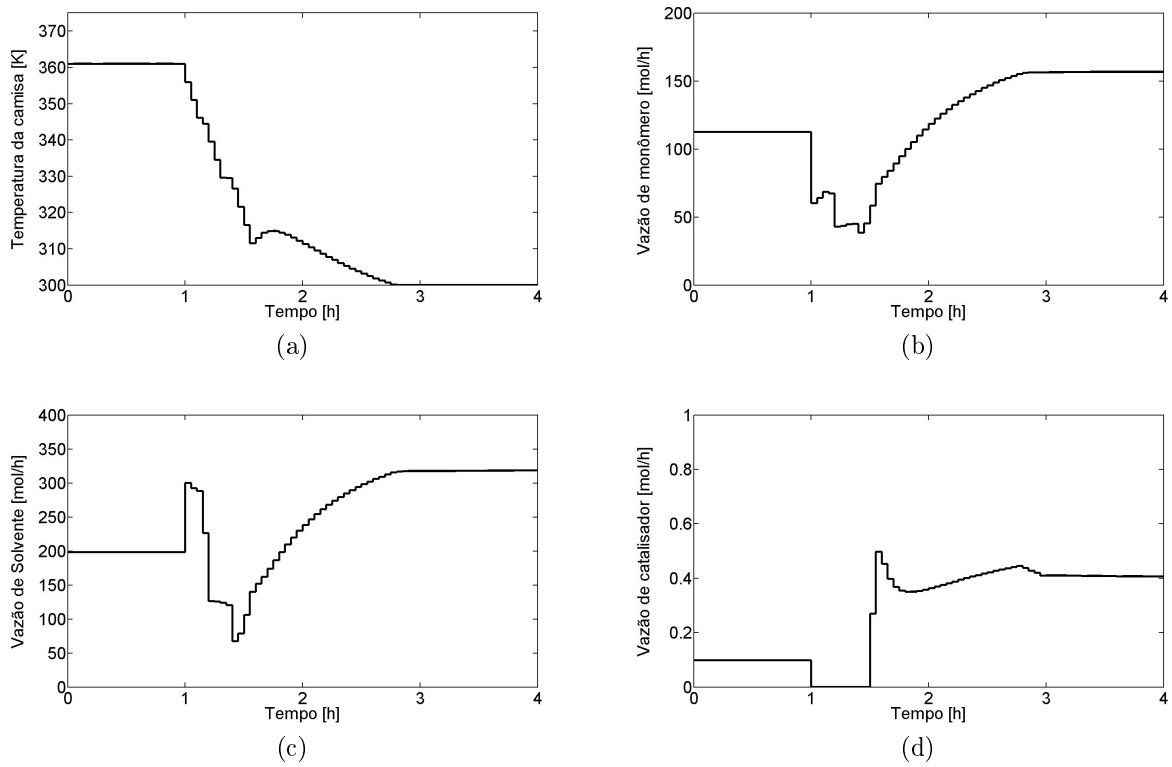


Figura 4.11: Desempenho do controlador MIMO: $Setpoint\ 1 \rightarrow Setpoint\ 2$. Primeira sintonia do peso Q . Variáveis manipuladas.

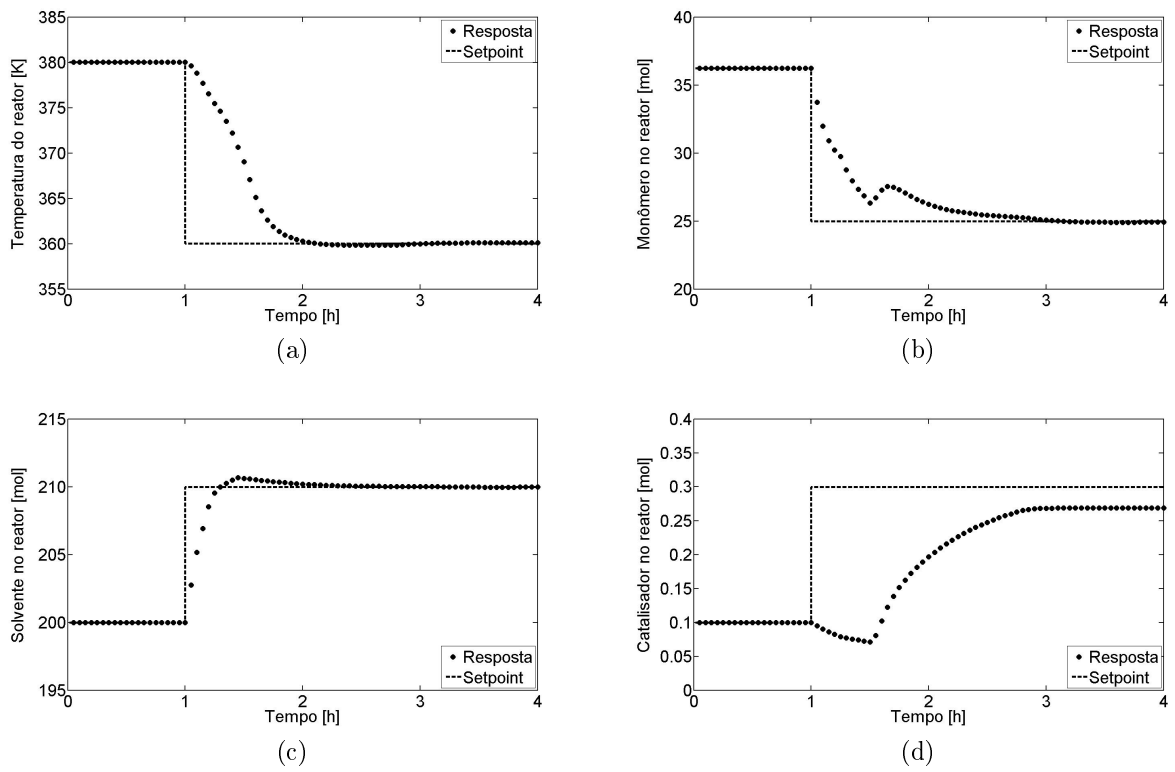


Figura 4.12: Desempenho do controlador MIMO: $Setpoint\ 1 \rightarrow Setpoint\ 2$. Segunda sintonia do peso Q . Variáveis controladas.

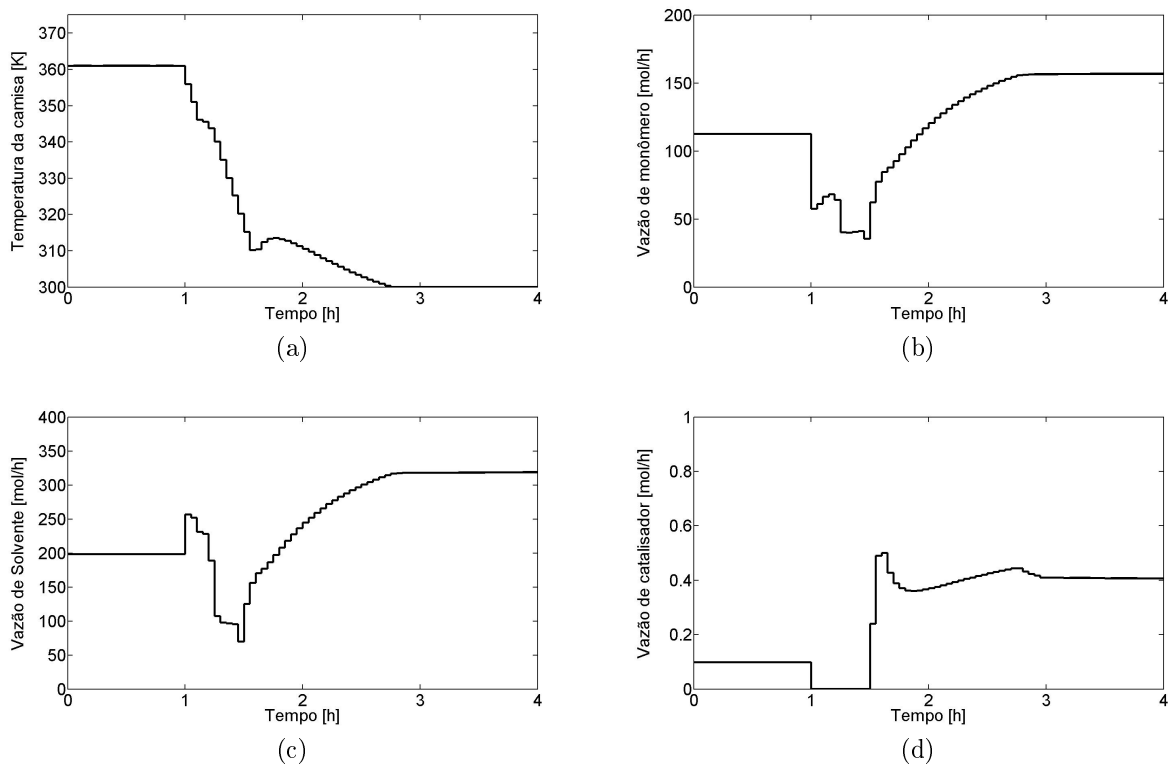


Figura 4.13: Desempenho do controlador MIMO: $Setpoint\ 1 \rightarrow Setpoint\ 2$. Segunda sintonia do peso Q . Variáveis manipuladas.

aumento da fração de catalisador provoca queda da fração de monômero, em virtude da reação química, causando um *offset* no *setpoint* do monômero. O aumento da taxa de reação, por sua vez, provoca aumento de temperatura, resultando em mais um *offset*. Na sintonia proposta, a influência do erro de *setpoint* do catalisador na função objetivo é menor que a influência dos erros do monômero e da temperatura, situação que pode ser revertida com o aumento do peso associado ao catalisador. As Figuras 4.14 e 4.15 trazem os resultados para um peso de 10000 referente ao catalisador, cem vezes superior ao valor inicial de 100. Desta vez, a quantidade de catalisador no reator segue desde o início para o *setpoint*, provocando, contudo, desvios na temperatura e no monômero e, ainda, *overshoot* no solvente (cujo peso atual é 1). Nota-se, portanto, que em cenários onde o *Setpoint 2* não é alcançável, o controlador tenta privilegiar algumas variáveis em detrimento de outras. Neste caso, temperatura e monômero devem ser privilegiados, visto que o aumento do peso associado ao catalisador prejudica o controle das demais variáveis, fato que ocorre em virtude de sua influência na taxa de reação. Além disso, na prática, o controle de temperatura e de conversão (produção) são muito mais importantes que o controle de catalisador (que pode, no entanto, afetar significativamente os custos do processo).

Todas as simulações realizadas até o momento levaram em consideração na função objetivo do controlador apenas os termos relacionados aos pesos \mathcal{Q} e \mathcal{R} . A única modificação dos pesos definidos na estratégia inicial de sintonia (Tabela 4.3) foi feita no peso referente ao solvente (M_{Sol}), associado ao parâmetro \mathcal{Q} , que passou de 0,5 para 1. Os valores iniciais definidos para compor o peso \mathcal{R} se mostraram satisfatórios. Resta, agora, definir os valores que compõem o peso \mathcal{S} . As Figuras 4.16 e 4.17 trazem os resultados da simulação considerando os valores iniciais (Tabela 4.3) definidos para a matriz \mathcal{S} . Esta simulação ilustra bem o comportamento discutido no início desta Seção (4.2). Note que as linhas tracejadas da Figura 4.16 indicam exatamente o mesmo *setpoint* utilizado em todas as simulações anteriores (*Setpoint 2*); contudo, desta vez, o controlador foi habilitado a perseguir também os valores estacionários das variáveis manipuladas, destacados na parte inferior da Tabela 4.2, como mostra a Figura 4.17. O resultado é um estado estacionário diferente do previsto para solvente e catalisador (Figuras 4.16c e 4.16d). A partir desta simulação, vê-se que a sintonia do parâmetro \mathcal{S} afeta o desempenho do sistema, como poderia ser esperado. A fim de enriquecer a análise, decidiu-se também utilizar um *setpoint* alcançável para as variáveis controladas, deslocando o sistema do *Setpoint 1* para o *Setpoint 3*, discriminado na Tabela 4.2.

As Figuras 4.18 e 4.19 mostram o desempenho do controlador sem a influência do segundo termo da Equação 3.16; ou seja, sem perseguir os *targets*. Como o novo *setpoint* é alcançável, não houve *offset* em nenhuma variável controlada.

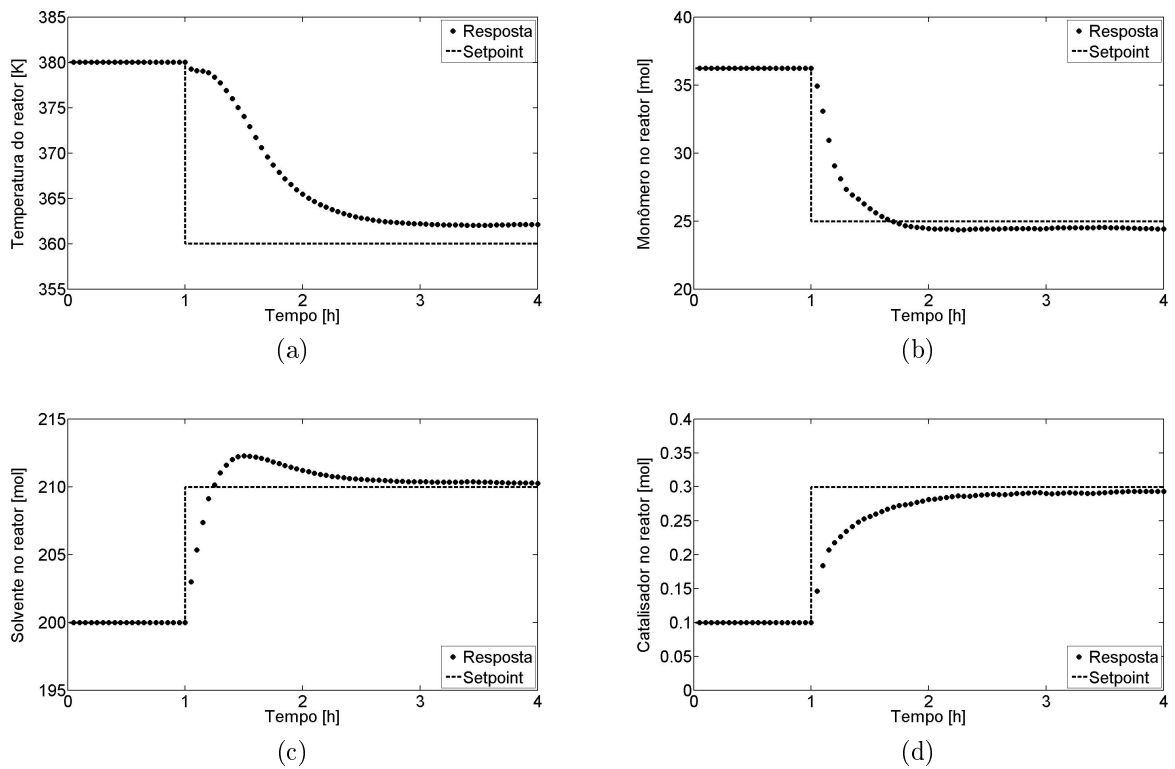


Figura 4.14: Desempenho do controlador MIMO: *Setpoint* 1 \rightarrow *Setpoint* 2. Terceira sintonia do peso \mathcal{Q} . Variáveis controladas.

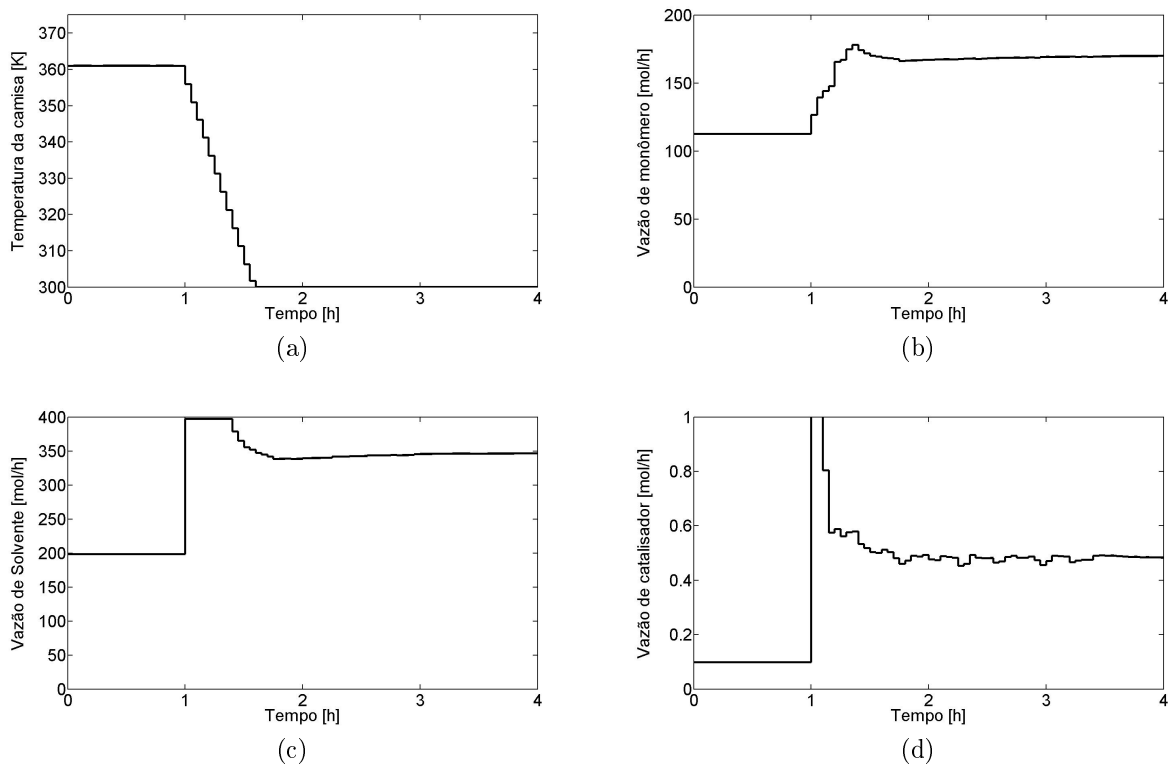


Figura 4.15: Desempenho do controlador MIMO: *Setpoint* 1 \rightarrow *Setpoint* 2. Terceira sintonia do peso \mathcal{Q} . Variáveis manipuladas.

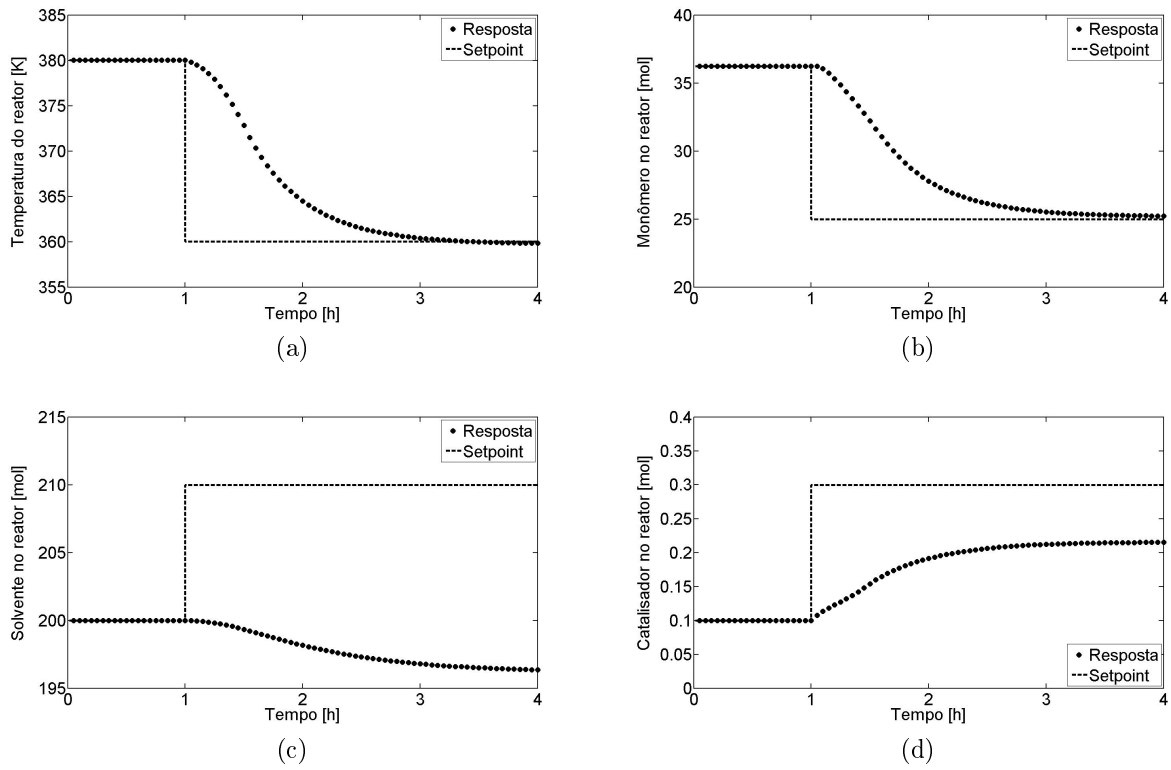


Figura 4.16: Desempenho do controlador MIMO: *Setpoint 1* \rightarrow *Setpoint 2*. Primeira sintonia do peso \mathcal{S} . Variáveis controladas.

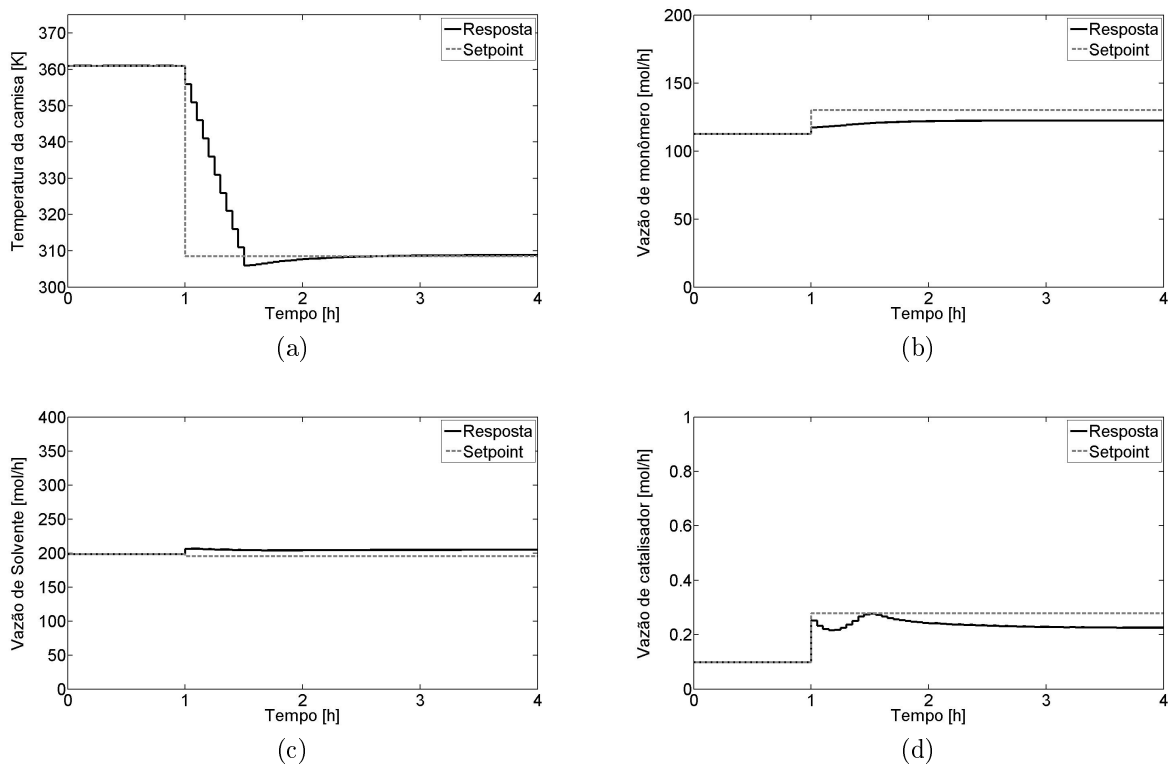


Figura 4.17: Desempenho do controlador MIMO: *Setpoint 1* \rightarrow *Setpoint 2*. Primeira sintonia do peso \mathcal{S} . Variáveis manipuladas.

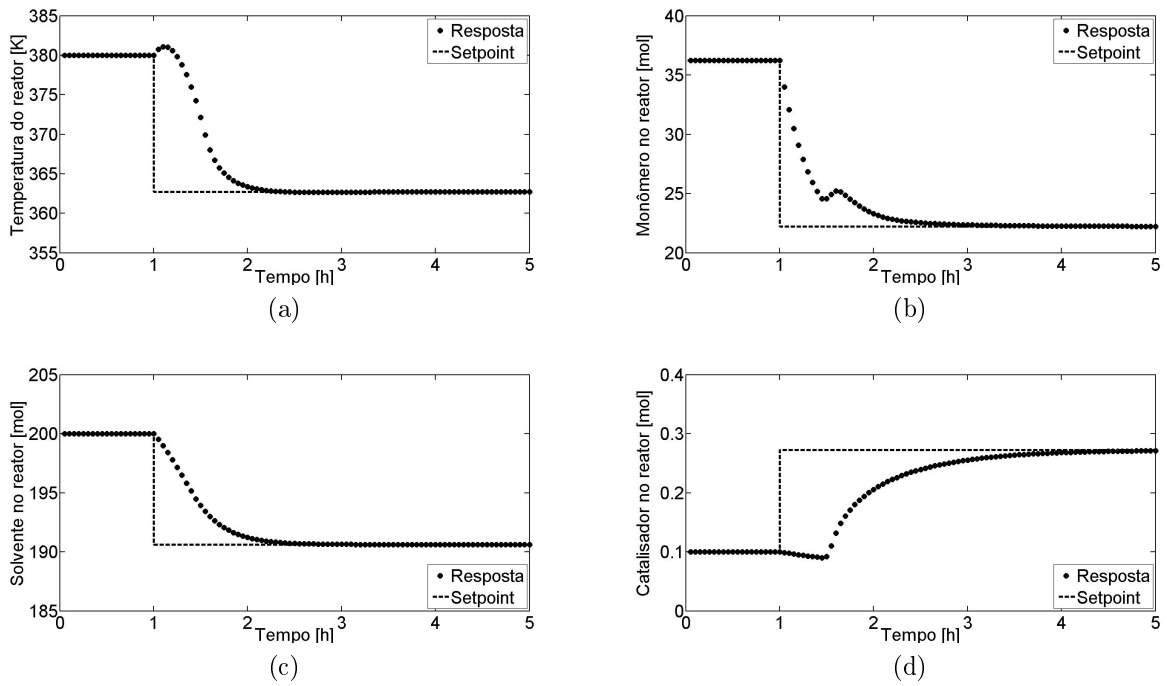


Figura 4.18: Desempenho do controlador MIMO: *Setpoint 1* \rightarrow *Setpoint 3*. Desconsiderando o termo associado ao peso \mathcal{S} . Variáveis controladas.

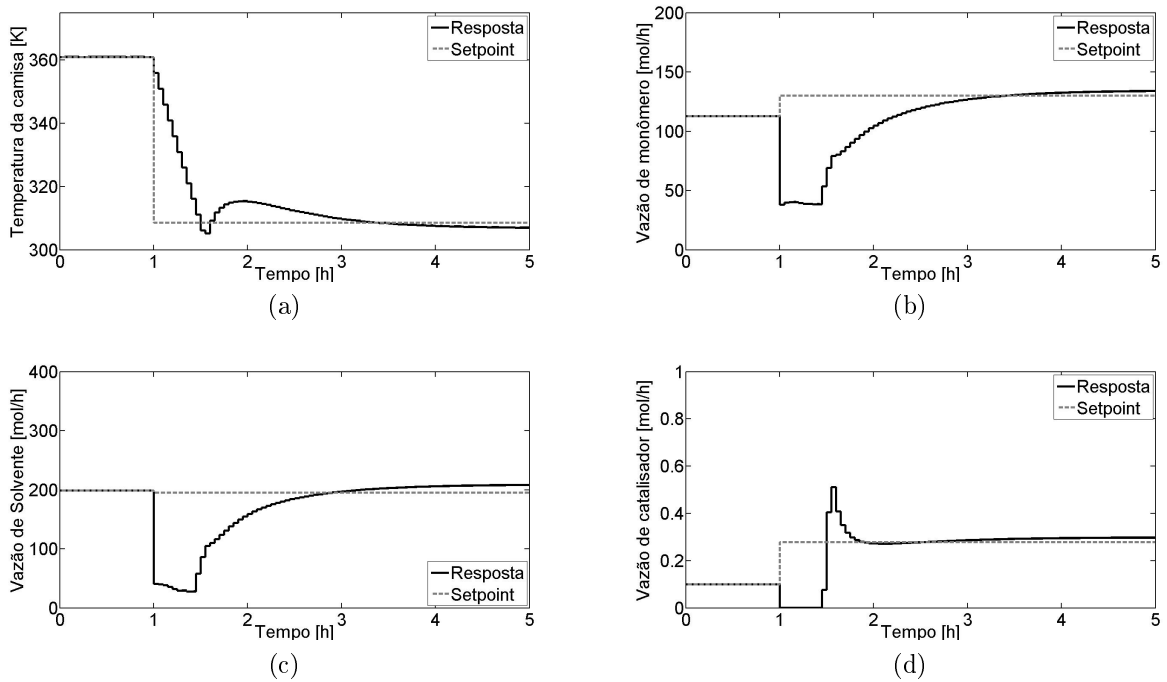


Figura 4.19: Desempenho do controlador MIMO: *Setpoint 1* \rightarrow *Setpoint 3*. Desconsiderando o termo associado ao peso \mathcal{S} . Variáveis manipuladas.

Os perfis das variáveis manipuladas se mostraram satisfatoriamente suaves; porém, observam-se pequenos desvios em relação aos valores estacionários, mas que não trazem problemas ao desempenho controlador.

Ativando o segundo termo da Equação 3.16 e aplicando a sintonia inicial para a matriz \mathcal{S} , obtêm-se os resultados mostrados nas Figuras 4.20 e 4.21. Os valores estacionários das variáveis manipuladas são rigorosamente seguidos, como mostrado na Figura 4.21; entretanto, observam-se desvios consideráveis nas variáveis controladas, fenômeno não observado na simulação que desconsidera os *targets*. Outras simulações foram realizadas com o intuito de diminuir a contribuição dos *targets* na função objetivo do controlador, mas não houve ganho de desempenho em relação às simulações sem a contribuição do referido termo.

Desempenho do controlador frente ao ruído

Em todas as simulações anteriores, foi considerado que os estados possuem medidas livres de incertezas. Nesta seção, são avaliados os impactos de incertezas nas medidas dos estados sobre o desempenho do controlador. Para tanto, as medidas dos estados foram corrompidas com ruídos normais de média (μ) zero. Para as variáveis M_{Mon} , M_{Pol} , M_{Sol} e M_{Cat} , o valor atribuído ao desvio padrão (σ) foi equivalente a 1% do valor medido. Para a temperatura do reator (T) o desvio padrão foi considerado constante e igual a 0,2 (que corresponde a cerca de 0,05% do valor medido). Tais valores são próximos aos encontrados em operações industriais, como se pode confirmar no trabalho de PRATA *et al.* (2009). A simplificação do desvio padrão correspondente à temperatura do reator não interfere significativamente nos resultados e foi admitida apenas para facilitar a discussão das análises posteriores.

Os resultados de desempenho do controlador SISO para uma mudança *Setpoint 1* \rightarrow *Setpoint 3* são mostrados na Figura 4.22. Observa-se que o ruído obteve efeito mais destacado para o monitoramento do solvente, do polímero e do monômero no reator. O efeito do ruído na temperatura foi pequeno e não prejudicou o cálculo das ações de controle. Apesar de alimentado com dados ruidosos, o integrador convergiu em todas as iterações.

Os resultados de desempenho do controlador MIMO para uma mudança *Setpoint 1* \rightarrow *Setpoint 3* são mostrados nas Figuras 4.23 e 4.24. Nota-se que o controlador consegue levar todas as variáveis para seus respectivos *setpoints*. Apenas o solvente apresenta uma dispersão acentuada, apesar do fato de estar sujeito às mesmas intensidades de ruídos que os outros componentes. Esta dispersão prejudica a determinação exata do estado da respectiva variável. As variáveis manipuladas apresentaram um comportamento mais agitado, como se poderia esperar. Mas de forma geral, o desempenho do controlador foi pouco influenciado pelo ruído.

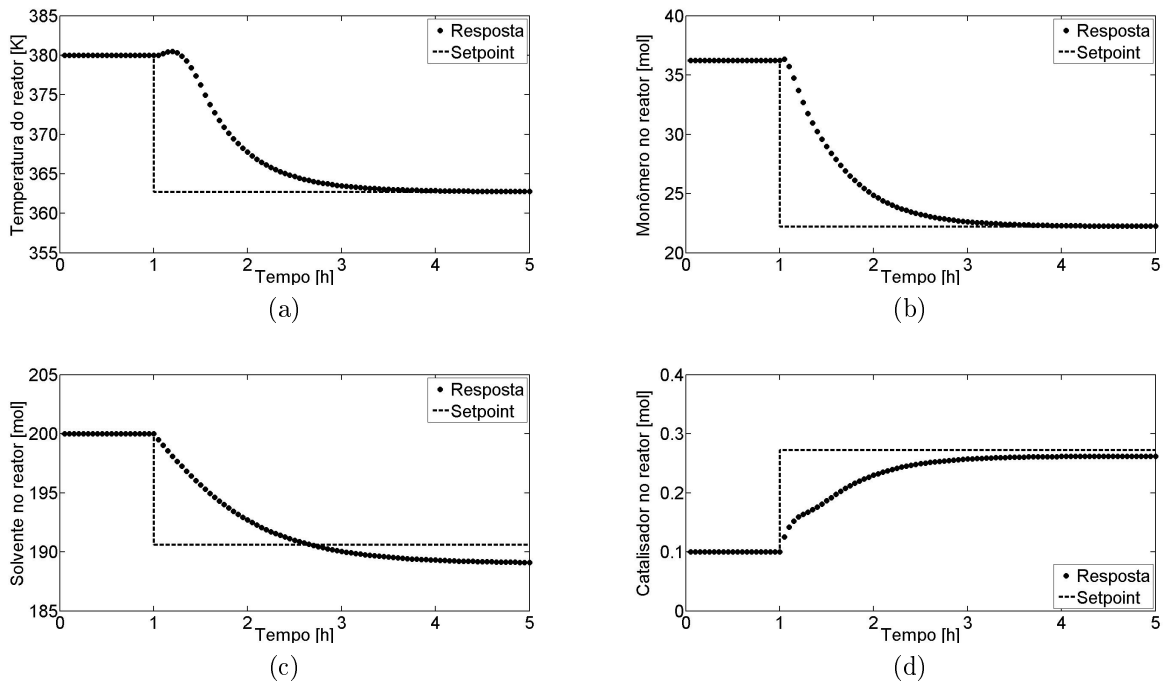


Figura 4.20: Desempenho do controlador MIMO: $Setpoint\ 1 \rightarrow Setpoint\ 3$. Primeira sintonia do peso \mathcal{S} . Variáveis controladas.

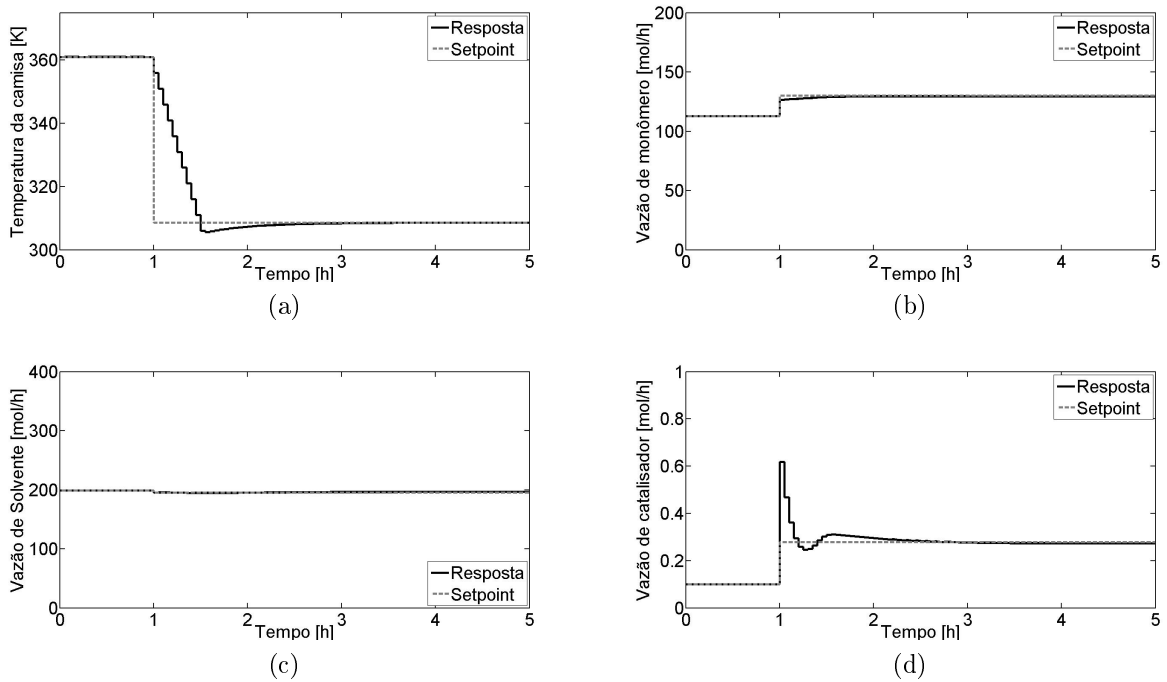


Figura 4.21: Desempenho do controlador MIMO: $Setpoint\ 1 \rightarrow Setpoint\ 3$. Primeira sintonia do peso \mathcal{S} . Variáveis manipuladas.

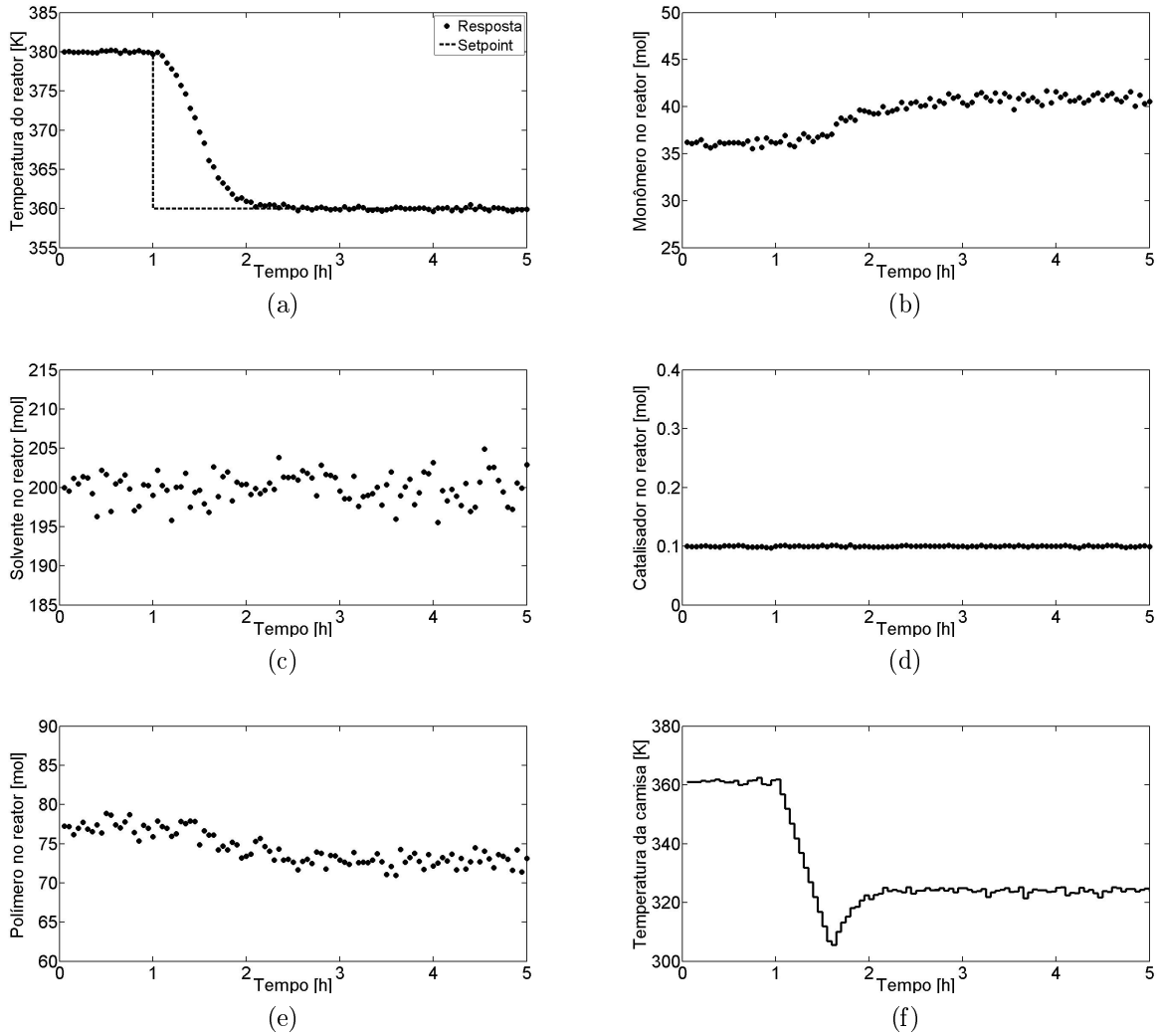


Figura 4.22: Desempenho do controlador SISO frente ao ruído: *Setpoint 1* \rightarrow *Setpoint 3*.

A configuração final da sintonia do controlador é dada pela Tabela 4.4. O segundo termo da Equação 3.16 não ofereceu melhoria no desempenho do controlador e, por isso, não será utilizado nas simulações posteriores. Sendo assim, o parâmetro \mathcal{S} foi removido da Tabela 4.4. Alterações nos horizontes de controle e previsão não produziram melhorias de desempenho, de forma que os valores iniciais foram mantidos.

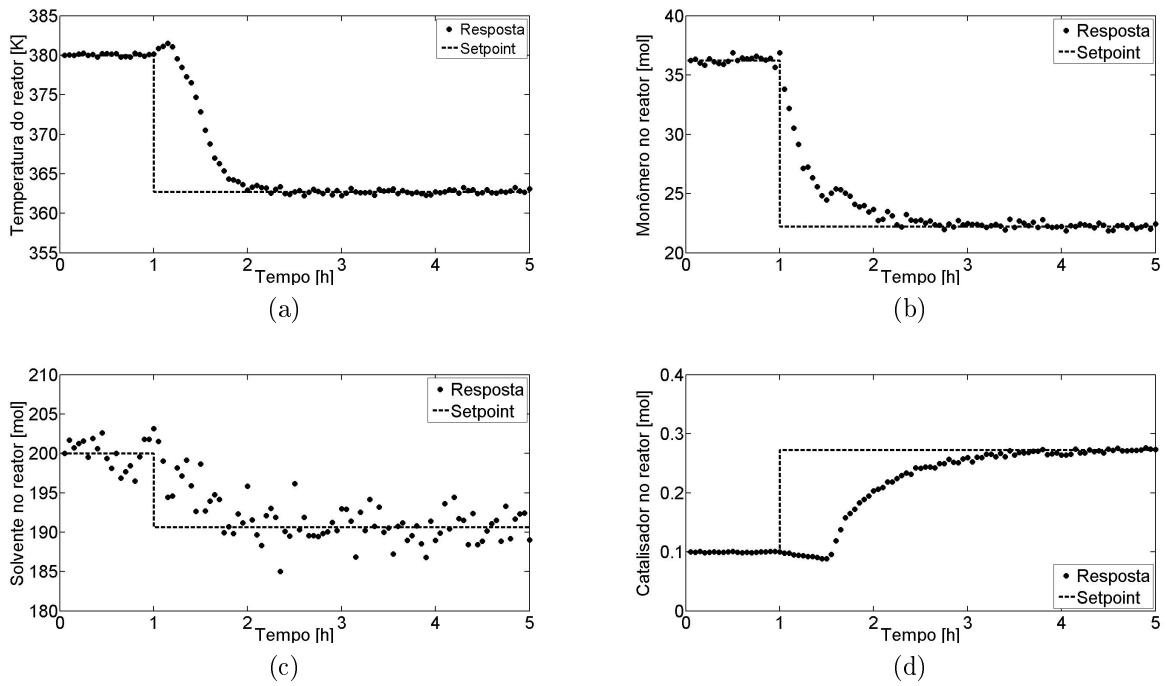


Figura 4.23: Desempenho do controlador MIMO frente ao ruído: *Setpoint 1* \rightarrow *Setpoint 3*. Variáveis controladas.

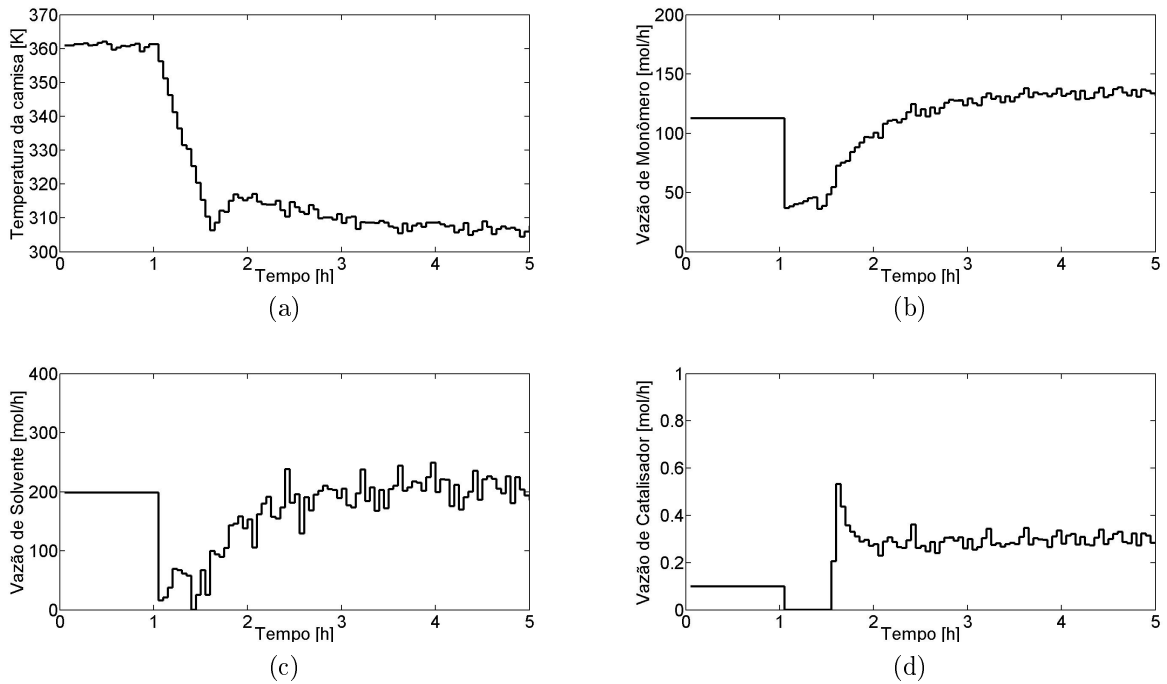


Figura 4.24: Desempenho do controlador MIMO frente ao ruído: *Setpoint 1* \rightarrow *Setpoint 3*. Variáveis manipuladas.

Tabela 4.4: Sintonia final do controlador.

Parâmetro	Var. associada	Valor
\mathcal{Q}	T	1
	M_{Mon}	1
	M_{Sol}	1
	M_{Cat}	100
\mathcal{R}	T_c	1
	F_{in}^{Mon}	0,5
	F_{in}^{Sol}	0,25
	F_{in}^{Cat}	100
M		3
P		10

4.3 Desempenho do reconciliador

Nesta seção, o desempenho do reconciliador é avaliado em função do horizonte de reconciliação H_{rec} e da intensidade do ruído adicionado à variável medida, neste caso, apenas o estado T . Os outros estados são estimados pelo algoritmo. As simulações conduzidas nesta etapa visam manter a variável controlada T no valor definidos pelo *Setpoint* 1 (Tabela 4.2), que é a condição inicial das simulações. O controlador atua na configuração SISO; isto é, apenas a temperatura da camisa é manipulada, as vazões de alimentação permanecem constantes. O parâmetro κ_c foi mantido constante e igual a 1 para $t \leq 1$. Para $t > 1$, são propostas três situações para o parâmetro κ_c : continua constante, redução degrau e redução exponencial (que simulam incrustação nas paredes do reator), cujos resultados são mostrados a seguir.

κ_c constante

O desempenho do algoritmo de reconciliação pode ser ajustado manipulando-se o horizonte de reconciliação H_{rec} . Para um valor inicial de 5, foram realizadas simulações com o objetivo de testar a sensibilidade da estimação frente aos ruídos $[\mu, \sigma]=[0; 0,2]$ e $[\mu, \sigma]=[0; 0,4]$. Os resultados desta simulação são mostrados na Figura 4.25, em que se pode notar o aumento na dispersão da variável estimada κ_c em função do aumento do ruído. Reduções em H_{rec} produzem estimações muito dispersas, prejudicando o desempenho do controlador. O resultado para o $H_{rec} = 8$ (Figura 4.26) mostra um desempenho bastante superior da trajetória de κ_c em relação à trajetória obtida com $H_{rec} = 5$ na condição de ruído $\sigma = 0,2$. O mesmo resultado é observado para a condição de ruído $\sigma = 0,4$, em que há um comportamento de tendência oscilatória para a trajetória de κ_c , que, ainda assim, é bastante

superior ao desempenho do reconciliador para $H_{rec} = 5$.

Nota-se que para ambos os tamanhos de H_{rec} houve perda de desempenho do controlador para aumentos na intensidade do ruído. Apesar de não haver interferência significativa na variável controlada T , houve um aumento da "agressividade" do controlador sobre a variável manipulada T_c . Destaque para a diferença no perfil da temperatura da camisa em cada tamanho do horizonte de reconciliação. Embora a magnitude do ruído influencie bastante a estimação do parâmetro κ_c , o efeito sobre a temperatura do reator foi pequeno em todos os casos simulados.

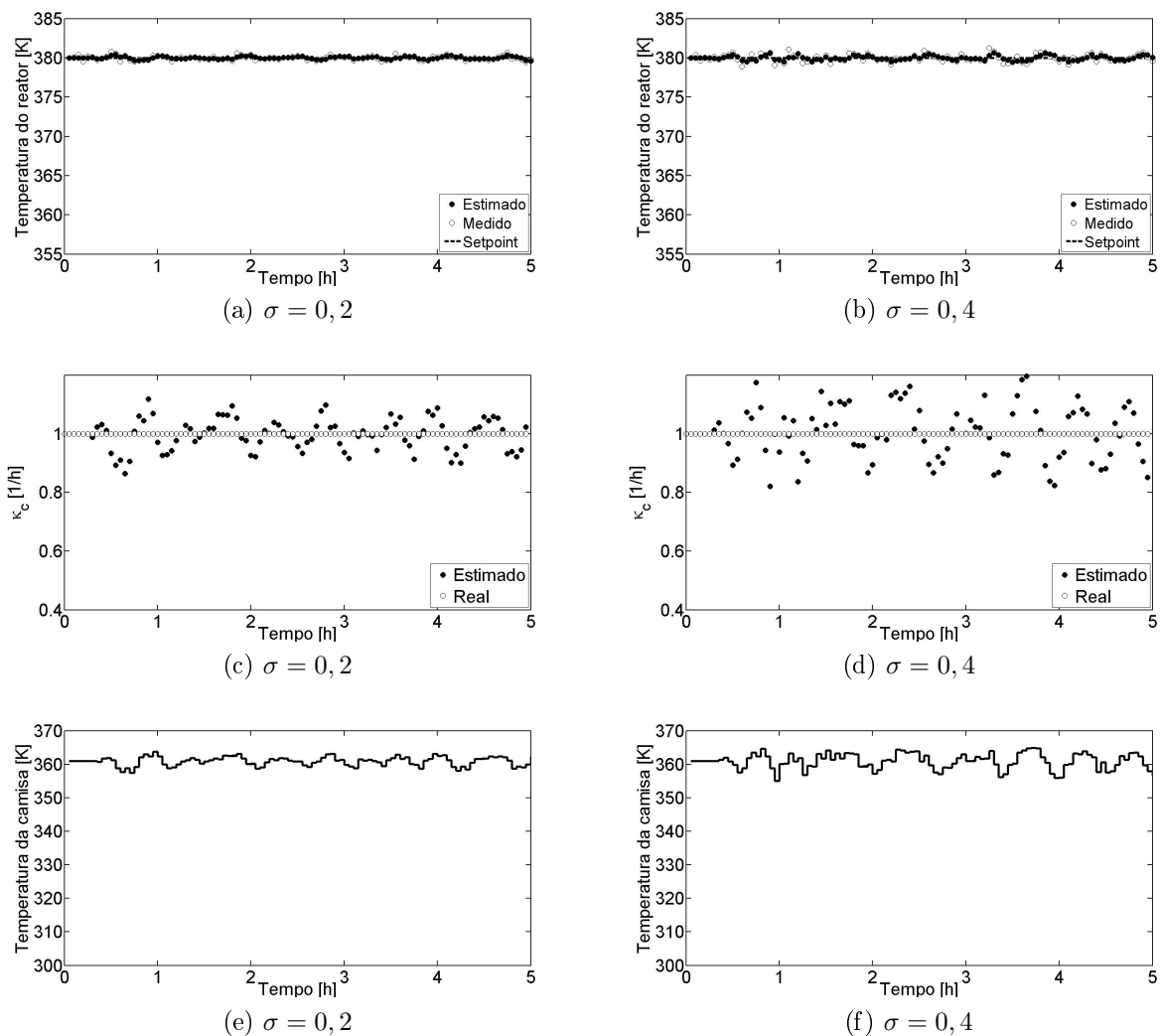


Figura 4.25: Desempenho do reconciliador frente ao ruído e κ_c constante: $H_{rec} = 5$.

κ_c variação degrau

Para uma variação degrau no parâmetro κ_c no instante 1 h, passando de 1 para 0,1, o desempenho do reconciliador para $H_{rec} = 8$ foi superior ao $H_{rec} = 5$ em ambas as condições de ruído, como pode se observar nas Figuras 4.27 e 4.28. Novamente, a

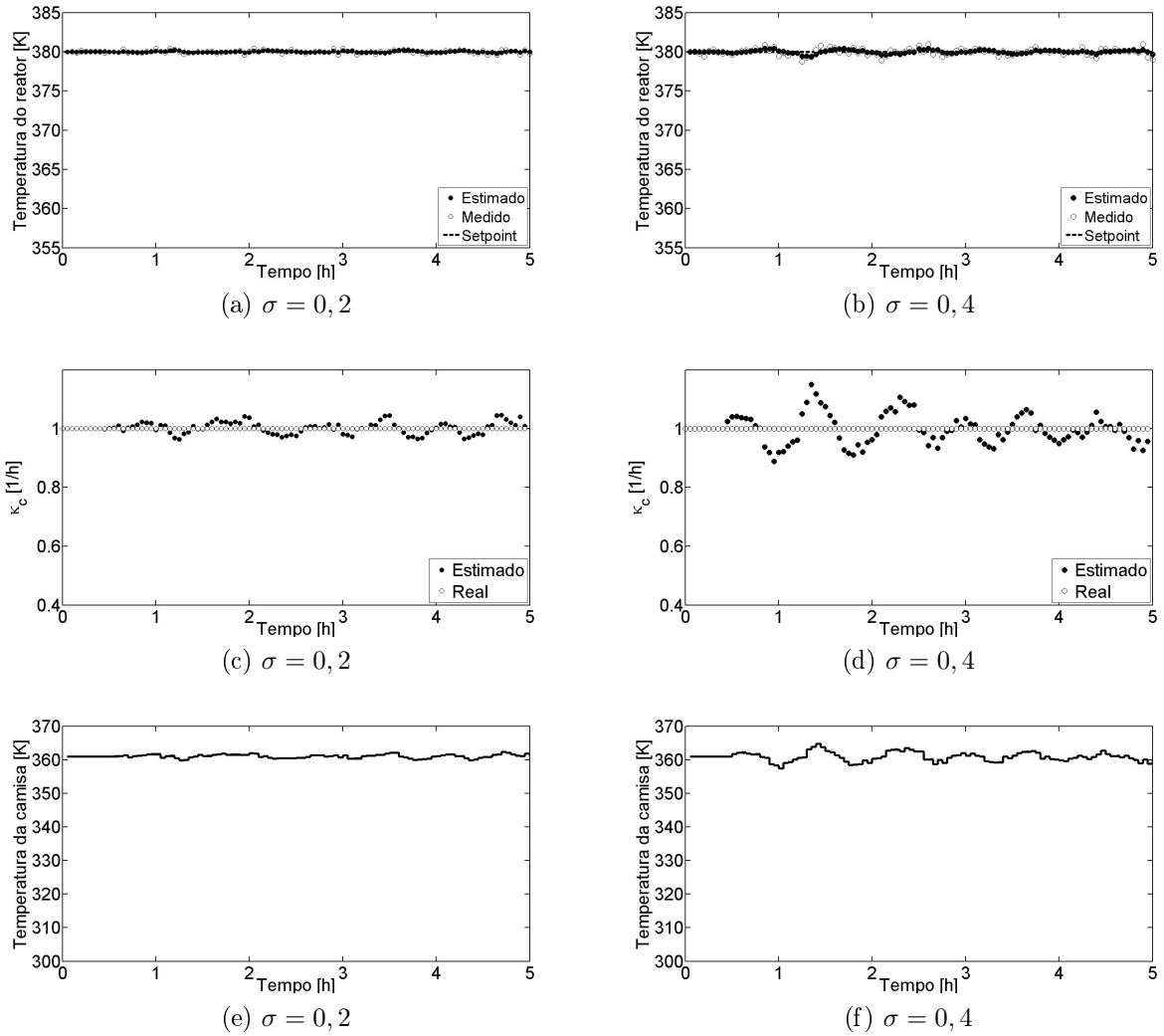


Figura 4.26: Desempenho do reconciliador frente ao ruído e κ_c constante: $H_{rec} = 8$.

estimação da temperatura do reator sofreu pouca influência do ruído e o perfil das ações de controle apresentou melhor desempenho para $H_{rec} = 8$.

κ_c variação exponencial

Para uma variação exponencial no parâmetro κ_c dada pela Equação 4.1 no instante 1 h, o desempenho do reconciliador para $H_{rec} = 8$, frente ao ruído $\sigma = 0,2$, foi ligeiramente superior ao $H_{rec} = 5$ em termos da estimação, mas se mostrou significativamente superior em termos do perfil da variável manipulada (Figuras 4.29 e 4.30). Para um ruído $\sigma = 0,4$, o maior horizonte apresenta melhor desempenho em todos os aspectos. Mais uma vez, não houve efeito significativo do ruído sobre a estimação da temperatura do reator, apenas uma leve dispersão para o ruído $\sigma = 0,4$

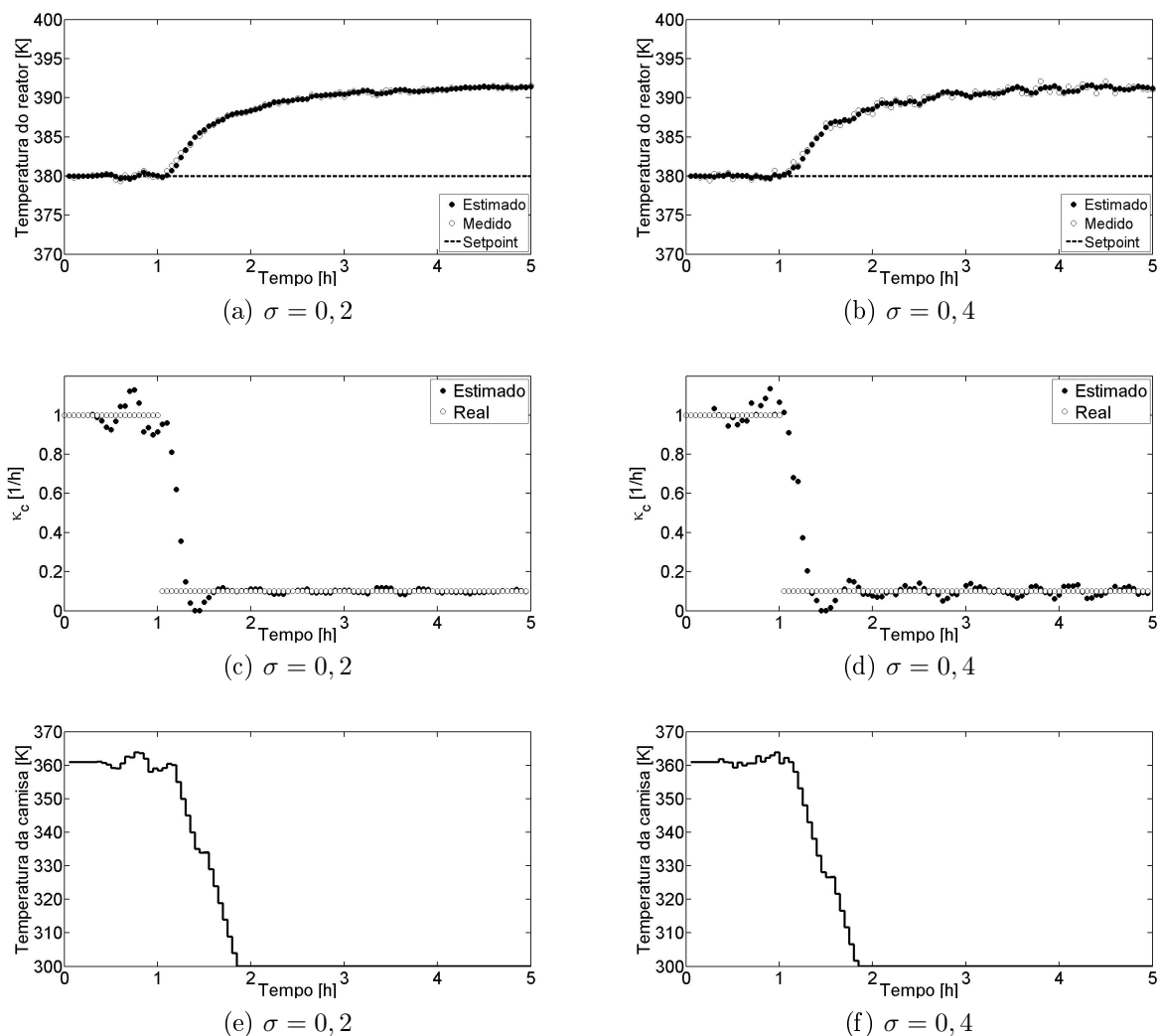


Figura 4.27: Desempenho do reconciliador frente ao ruído e variação degrau no κ_c : $H_{rec} = 5$.

e $H_{rec} = 5$ (Figura 4.29b), mas que não compromete a qualidade da estimação.

$$\kappa_c(t) = 0,1 + 0,9 \exp(-(t - 1)) \quad t > 1 \quad (4.1)$$

O ruído utilizado nas próximas simulações será de $\sigma = 0,2$ e $\mu = 0$, e o horizonte de reconciliação escolhido foi $H_{rec} = 8$, devido ao seu desempenho superior sem aumento significativo do custo computacional. O tempo médio gasto pelo reconciliador em cada iteração é de menos de 2 segundos.

Desempenho do controlador frente ao ruído considerando o reconciliador

Para validar as simulações deste trabalho em situações mais típicas do meio industrial, o comportamento do controlador foi testado na presença de ruído no

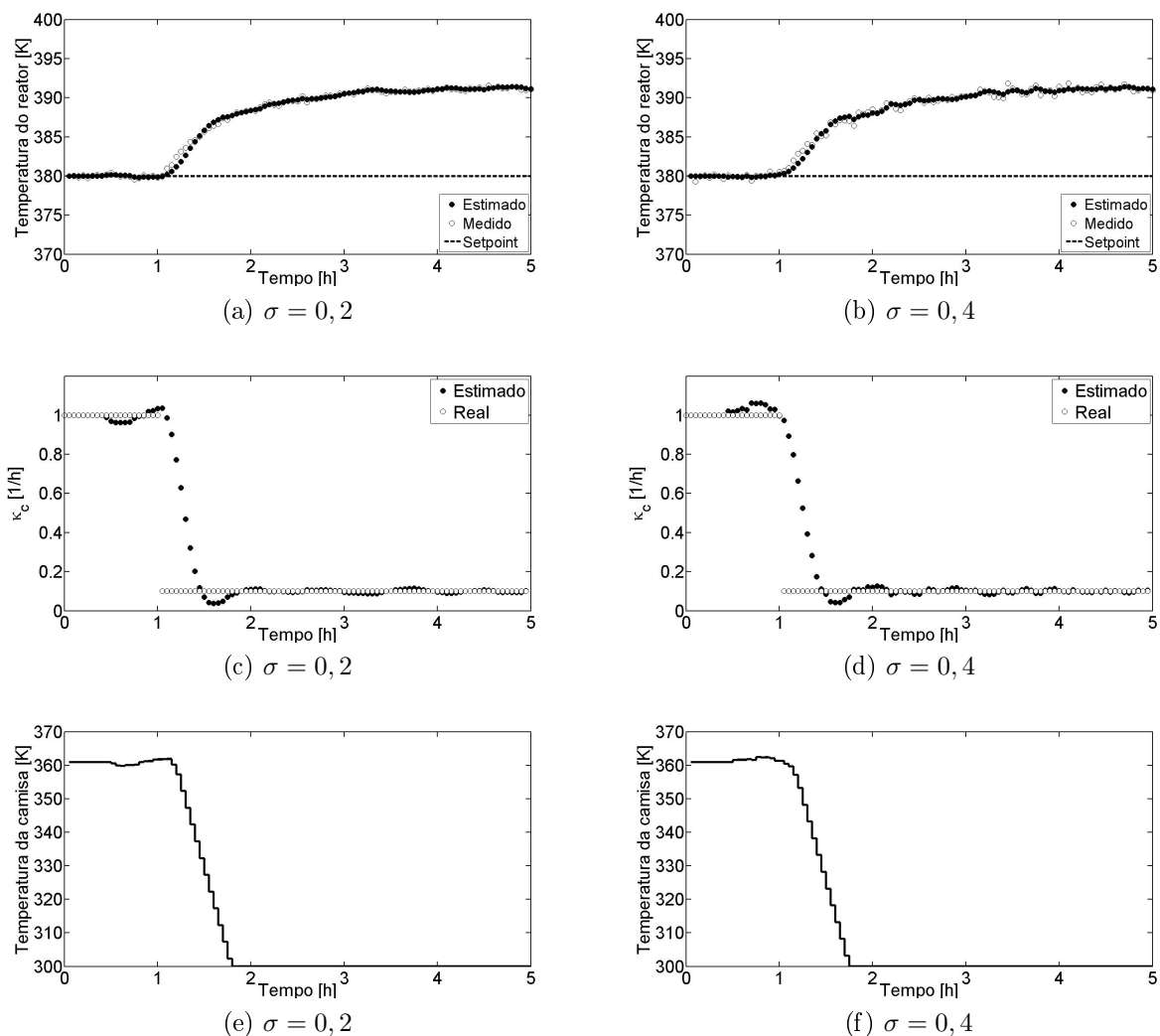


Figura 4.28: Desempenho do reconciliador frente ao ruído e variação degrau no κ_c : $H_{rec} = 8$.

sinal de controle. Nas simulações anteriores, discutidas na Seção *Desempenho do Controlador*, foi admitido que todos os estados eram conhecidos. Agora, a única variável medida é a temperatura do reator (T), como usualmente ocorre em processos reais, que foi corrompida com ruído normal de média (μ) zero e desvio padrão (σ) 0,2. Os outros quatro estados são estimados pelo reconciliador, concomitantemente à constante de troca térmica (κ_c). Nas simulações, o sistema parte do *Setpoint* 1 para o *Setpoint* 3. O κ_c do reator permanece constante com valor 1. Os resultados do controlador SISO são mostrados na Figura 4.31. As Figuras 4.32 e 4.33 trazem os resultados do controlador MIMO. Nota-se que os desempenhos dos controladores SISO e MIMO são bastantes semelhantes e não foram prejudicados pelo ruído de desvio padrão (σ) 0,2. Nem mesmo um ruído de desvio padrão (σ) 0,4 foi suficiente para prejudicar significativamente o desempenho do controlador, como pode ser observado nas Figuras 4.34 e 4.35. Portanto, a configuração e a sintonia do sistema

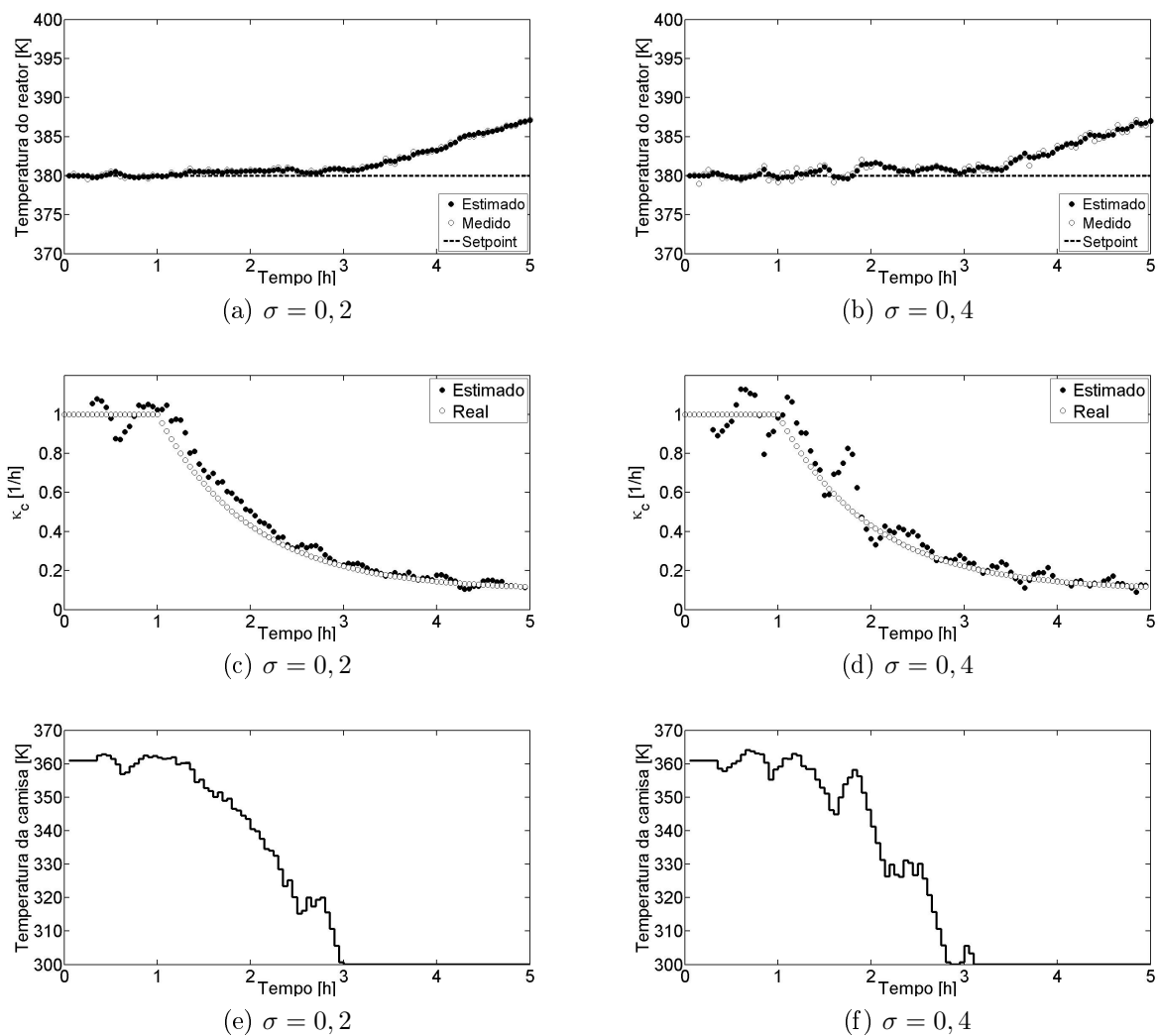
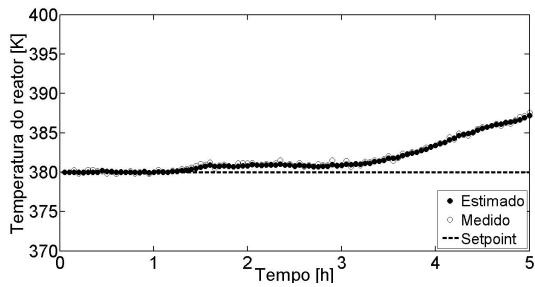
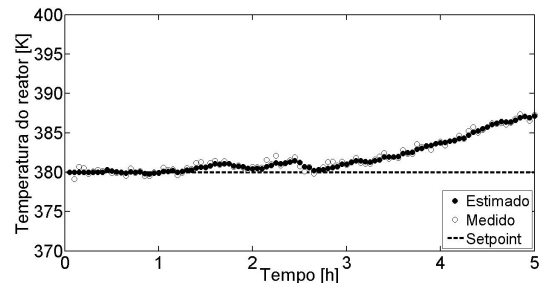


Figura 4.29: Desempenho do reconciliador frente ao ruído e variação exponencial no κ_c : $H_{rec} = 5$.

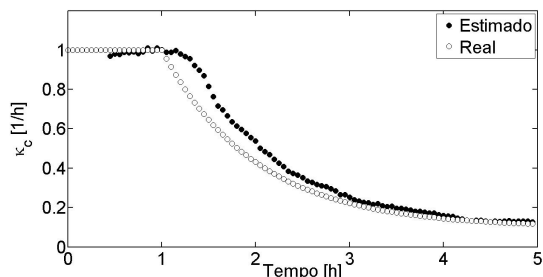
de controle e reconciliação se mostraram desempenhos satisfatórios nos testes aos quais foram submetidos. O tempo total gasto pelo reconciliador e pelo controlador em cada iteração varia de 2 a 20 segundos.



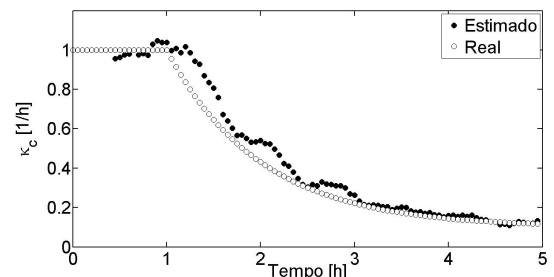
(a) $\sigma = 0,2$



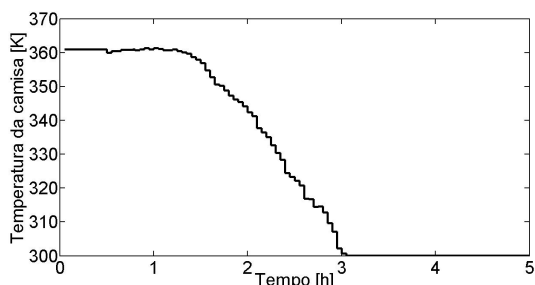
(b) $\sigma = 0,4$



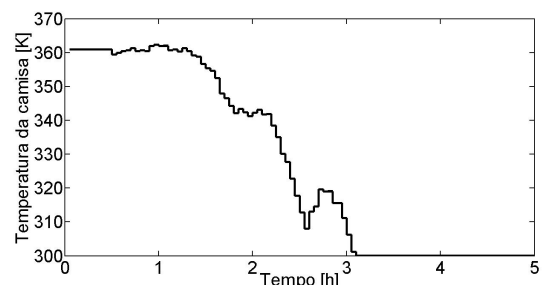
(c) $\sigma = 0,2$



(d) $\sigma = 0,4$



(e) $\sigma = 0,2$



(f) $\sigma = 0,4$

Figura 4.30: Desempenho do reconciliador frente ao ruído e variação exponencial no κ_c : $H_{rec} = 8$.

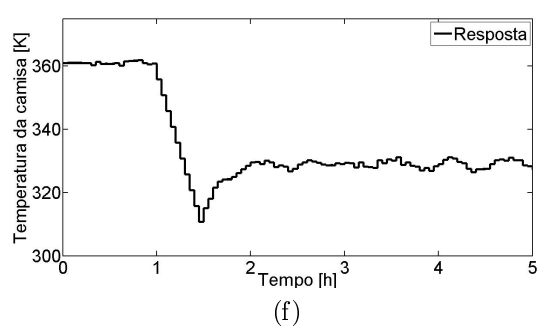
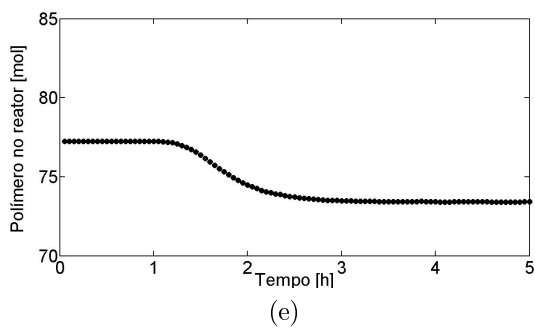
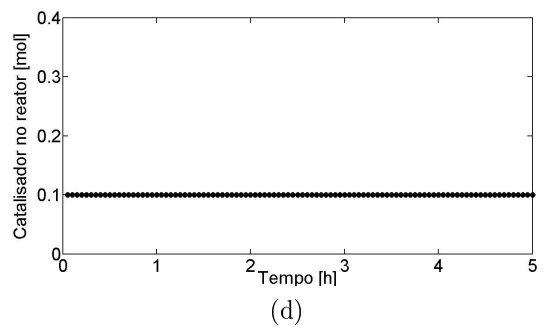
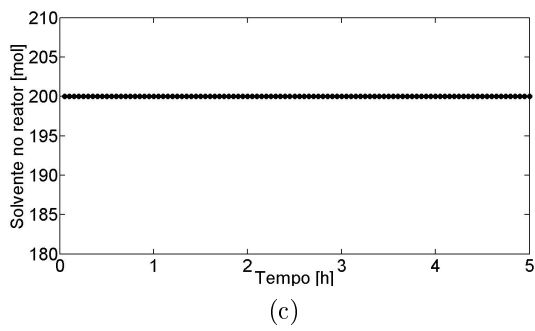
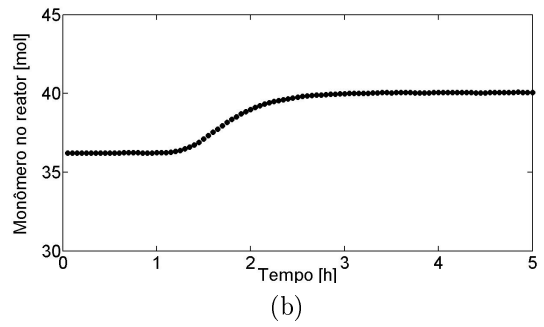
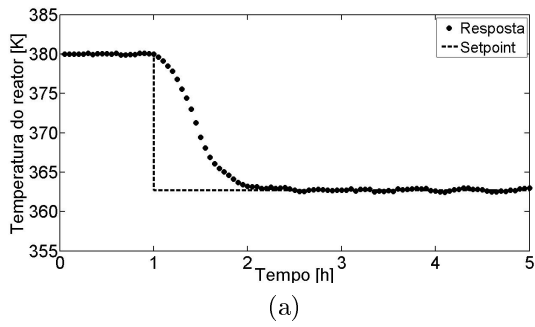


Figura 4.31: Desempenho do controlador SISO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,2]$.

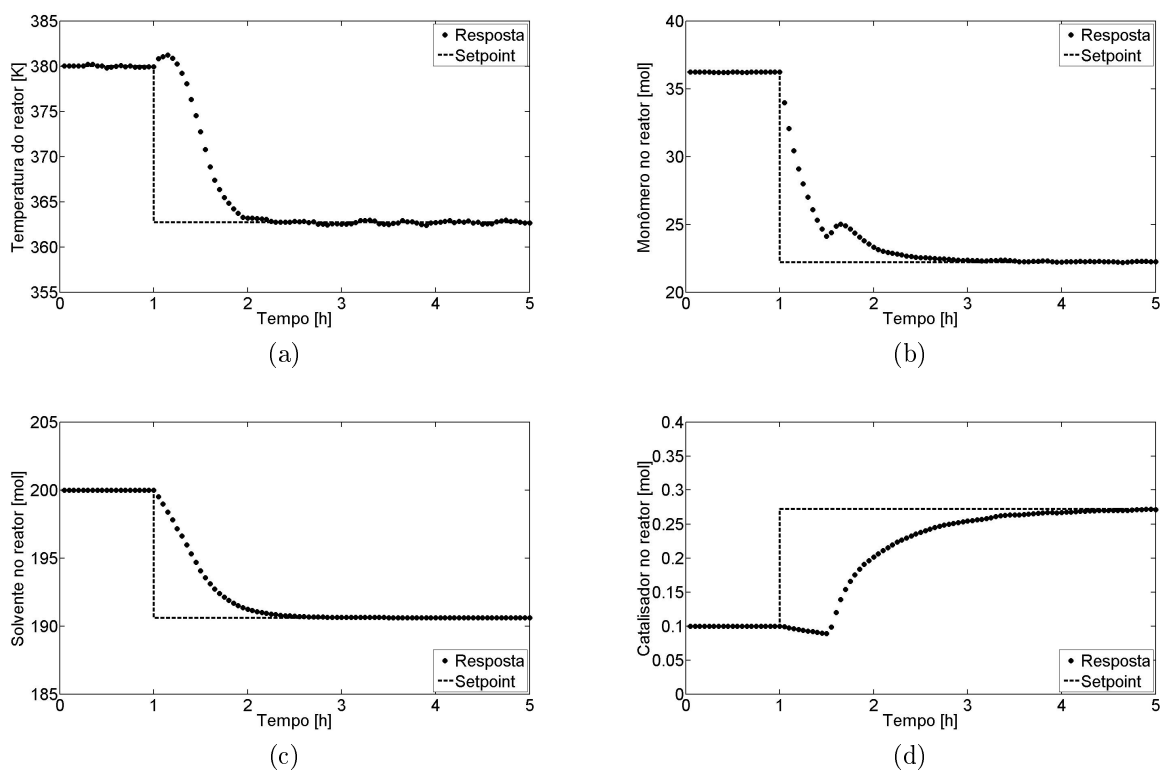


Figura 4.32: Desempenho do controlador MIMO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,2]$. Variáveis Controladas.

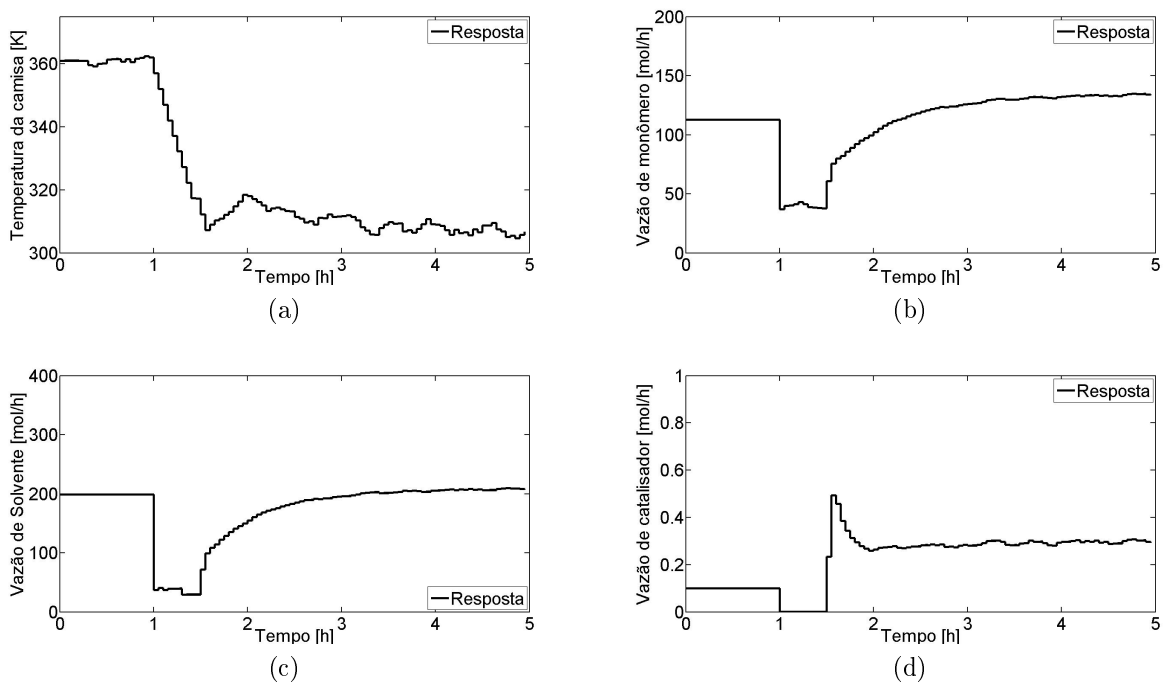


Figura 4.33: Desempenho do controlador MIMO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,2]$. Variáveis Manipuladas.

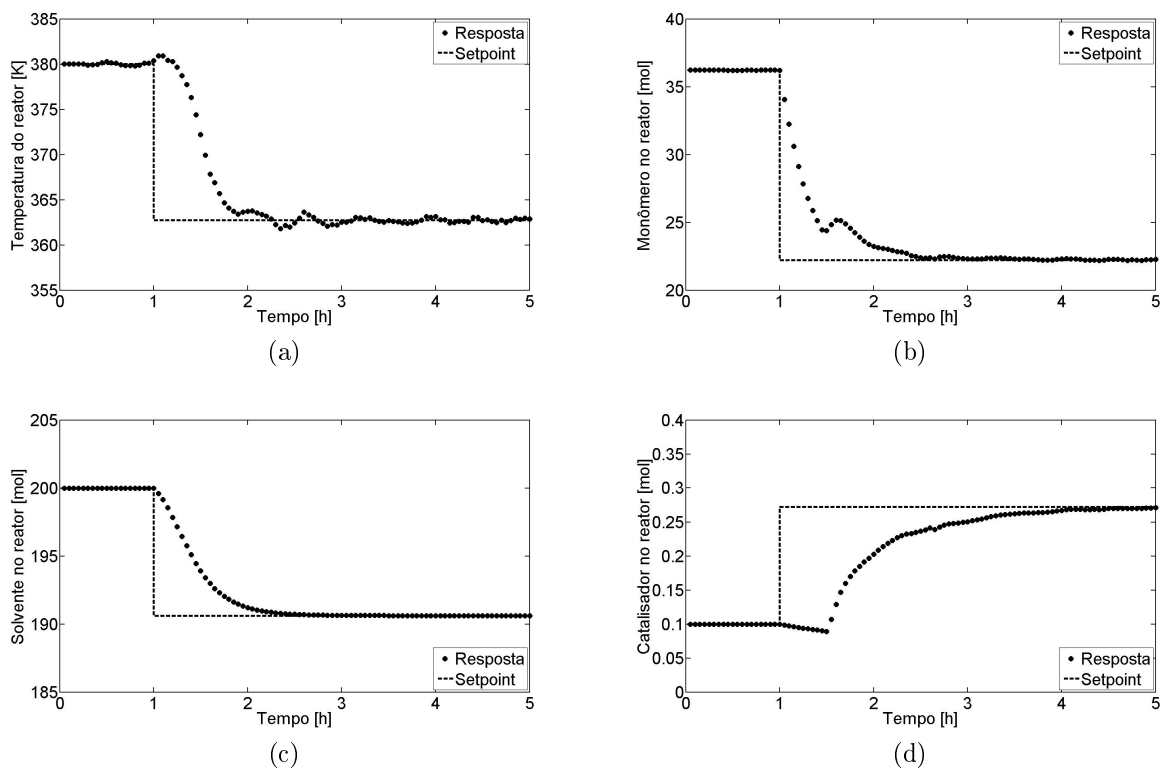


Figura 4.34: Desempenho do controlador MIMO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,4]$. Variáveis Controladas.

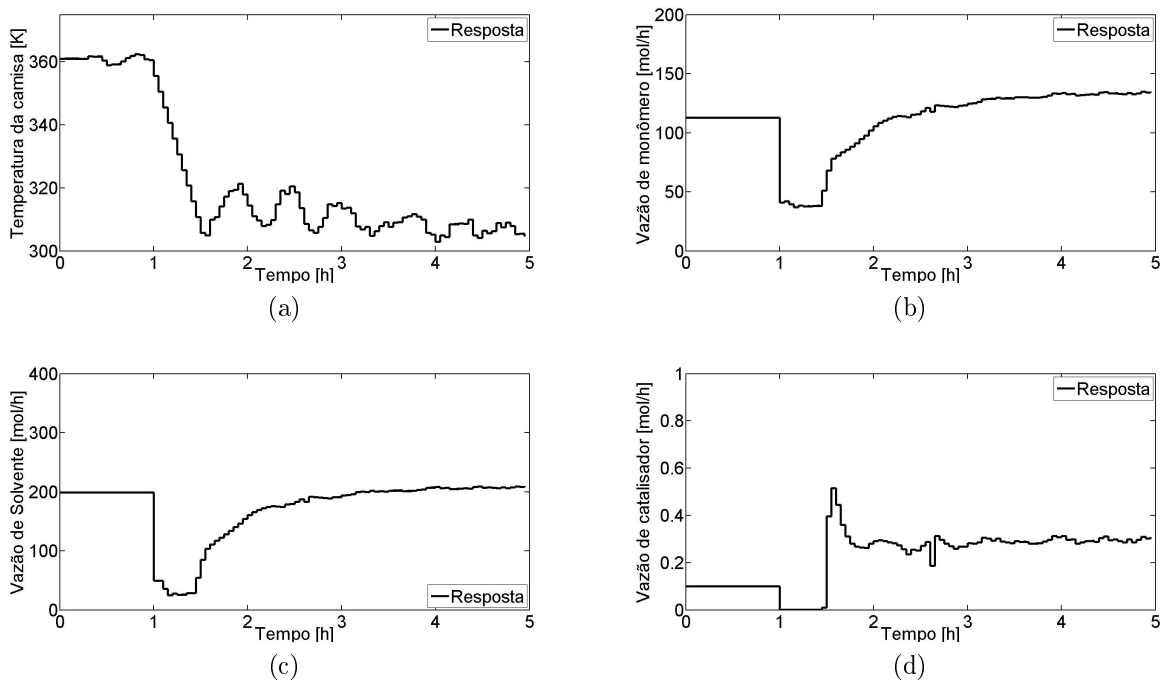


Figura 4.35: Desempenho do controlador MIMO frente ao ruído com reconciliação de dados: $[\mu, \sigma]=[0; 0,4]$. Variáveis Manipuladas.

4.4 Simulações de comportamentos inseguros

Com o objetivo de testar a eficiência do algoritmo de segurança desenvolvido nesta dissertação foram considerados alguns cenários nos quais eventos indesejados foram simulados. São eles:

- ☞ Perda da capacidade de refrigeração do reator;
- ☞ Adição de excesso de catalisador;
- ☞ Perda de controle da temperatura da camisa.

O limite de segurança considerado nas simulações foi referente à temperatura do reator, que deve atingir, no máximo, o valor de 385 K. Os eventos capazes de superar este valor são considerados cenários de risco e, portanto, ativam o sistema de segurança. O referido limite de segurança foi arbitrado com o objetivo de ilustrar o procedimento proposto neste trabalho. As simulações desta seção são realizadas no modo SISO.

4.4.1 Perda da capacidade de refrigeração do reator

A perda da capacidade de refrigeração é simulada através da variação do parâmetro κ_c . Este tipo de evento é bastante comum durante a operação de reatores com camisa. Diversas falhas podem provocar uma diminuição da capacidade de troca térmica entre camisa e reator, por exemplo, falha na bomba de alimentação de fluido, falha na agitação do reator, aumento da viscosidade da mistura reacional (efeito gel) e incrustação. Como o parâmetro κ_c não pode ser medido, ele deve estimado em tempo real, de maneira a acompanhar qualquer mudança indesejada durante a operação (detalhes do estimador na Seção 3.4). As falhas podem gerar variações bruscas e variações graduais no κ_c da camisa, que são simuladas como variações degrau e exponencial, respectivamente.

Reduções degrau no κ_c para valores entre 1 e 0,2 não representam risco para o processo. Apenas eventos que reduzam o parâmetro a valores menores que 0,2 são capazes de violar o limite de segurança de 385 K, como pode ser observado na Figura 4.36. O mesmo ocorre para reduções exponenciais no parâmetro (Figura 4.37). Observa-se na Figura 4.38 que o problema de troca térmica provoca a saturação da camisa, mas não evita a violação do limite de segurança. Portanto, reduções do κ_c para valores menores que 0,2 são eventos de risco potencial e devem ser evitadas.

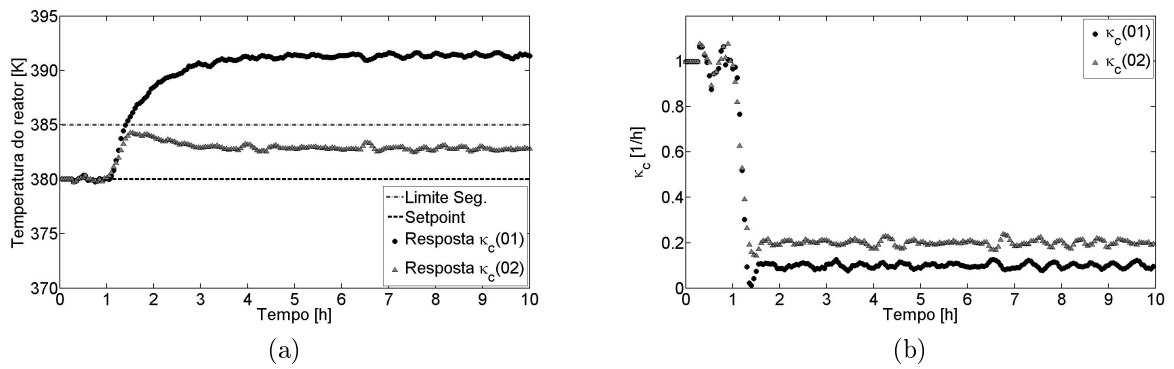


Figura 4.36: Efeito da redução brusca do κ_c .

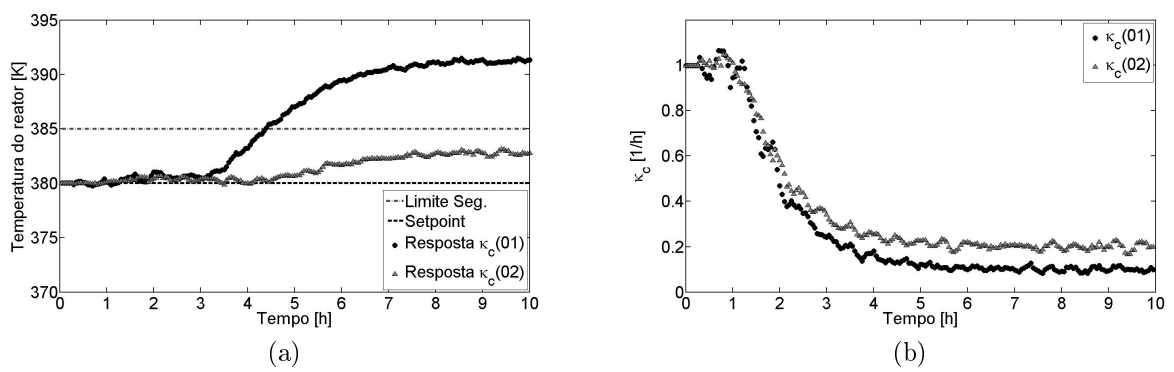


Figura 4.37: Efeito da redução gradual do κ_c .

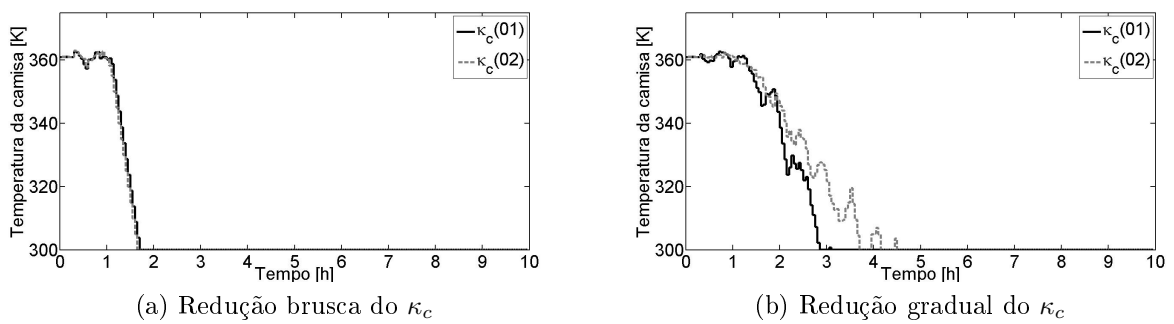


Figura 4.38: Perfil da variável manipulada T_c para os cenários de variação do κ_c .

4.4.2 Adição de excesso de catalisador

A adição de excesso de catalisador é simulada através da variação da vazão de alimentação de catalisador (F_{in}^{Cat}). É um evento tipicamente provocado por falha na bomba de catalisador ou ainda por falha no medidor de vazão, que poderia indicar um valor diferente do efetivamente alimentado. Este tipo de evento é bastante perigoso porque aumenta a taxa de reação e, conseqüentemente, pode provocar um *runaway*. Vale ressaltar que variações de atividade são comuns na planta, em função da variabilidade do processo de produção do catalisador e da presença de contaminantes.

Aumentos bruscos na vazão de catalisador para valores entre 0,1 (valor de F_{in}^{Cat} no *Setpoint* 1) e 0,9 não conduzem a situações de risco. Valores maiores que 0,9, no entanto, levam à violação do limite de temperatura, como mostra a Figura 4.39, e devem ser evitados. Não há perda de controle, mas a refrigeração não é rápida o suficiente para evitar que o limite de segurança seja violado.

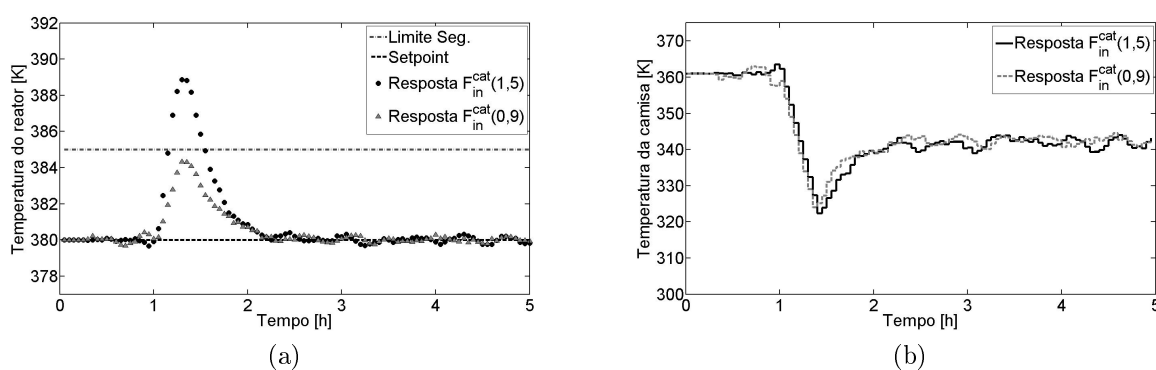


Figura 4.39: Efeito do aumento brusco de F_{in}^{Cat} .

4.4.3 Perda de controle da temperatura da camisa

A perda de controle da temperatura da camisa também é um evento comum durante a operação. Ocorre por falha no trocador de calor responsável pela manutenção da temperatura do fluido térmico ou por falha na geração de vapor. Como prejudica seriamente a capacidade de resfriamento do reator, pode provocar sérios aumentos de temperatura. Para o caso abordado neste trabalho, a temperatura da camisa é a única variável manipulada para o controle da temperatura do reator em condições normais de operação (configuração SISO), portanto, o travamento desta temperatura significa a total impotência do sistema de controle automático.

A zona de risco para aumentos bruscos de T_c surge para valores maiores que 370 K. Valores menores que este não representam risco ao processo, como ilustra a Figura 4.40.

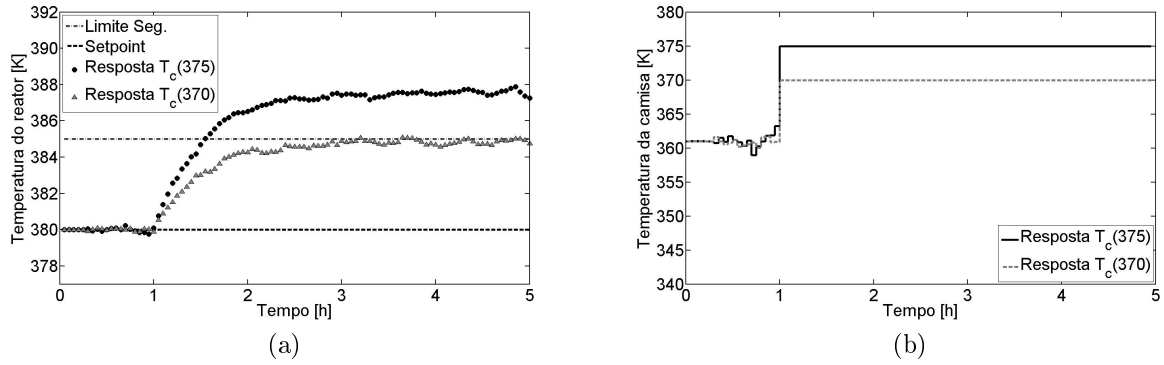


Figura 4.40: Efeito do aumento brusco de T_c .

4.5 O sistema integrado de segurança

As simulações partem de um estado estacionário definido pelo *Setpoint* 1 e têm seu risco avaliado para uma determinada falha no tempo de 1 h. De 1h até 4 h, o controlador atua na configuração MIMO com o objetivo de levar a planta para o *setpoint* de segurança (SP_{seg}). No tempo de 4 h o controlador volta para a configuração SISO e o evento anormal simulado no instante 1 h é efetivamente aplicado na planta, para testar a eficácia do sistema de segurança.

Os gráficos de variáveis controladas (VC), variáveis manipuladas (VM) e κ_c estimado são organizadas da seguinte maneira:

- ☞ A resposta da planta é representada por círculos pretos (VC e κ_c) e por linhas contínuas pretas (VM);
- ☞ A resposta do modelo na simulação da falha é representada por xis cinzas (VC e κ_c) e por linhas tracejadas com xis cinzas (VM);
- ☞ A resposta predita para a planta em situação de falha a partir do estado seguro calculado é representada por diamantes cinzas (VC e κ_c) e por linhas cinzas (VM). Esta trajetória é calculada no instante 1 h, mas foi deslocada para o instante 4 h com o objetivo de facilitar a comparação com o comportamento efetivo da planta;
- ☞ O estado seguro calculado é representado por uma linha tracejada (VC);
- ☞ O limite de segurança é representado por uma linha contínua (VC).

Os valores da matriz de pesos \mathcal{X} , utilizada na função objetivo para cálculo dos estados seguros (Equação 3.26), são idênticos aos da matriz \mathcal{Q} utilizada na função objetivo do controlador (Equação 3.16) e podem ser encontrados na Tabela 4.4.

4.5.1 Perda da capacidade de refrigeração

Nesta seção, o desempenho do sistema integrado de segurança é testado para o evento indesejado discutido na Seção 4.4.1, que evidencia os impactos provocados por variações no parâmetro κ_c no processo. O evento selecionado para avaliação do sistema de segurança foi a redução de grau na constante de troca térmica de 1 para 0,1, cuja dinâmica é mostrada na Figura 4.36 em círculos pretos. Os resultados são mostrados nas Figuras 4.41 e 4.42.

Como esperado, a simulação da falha prevê uma violação do limite de segurança (Figura 4.41a) e, imediatamente, ativa o buscador de estados seguros que, por sua vez, indica como seguro o *setpoint* representado pela linha tracejada. O controlador é reconfigurado para o modo MIMO e leva o sistema para o estado seguro. Note que o novo estado é próximo ao anterior. As mudanças mais acentuadas foram o aumento de 38% na quantidade de monômero (Figura 4.41b) e a redução de 18% na quantidade de polímero no reator (Figura 4.41e). O reconciliador conseguiu acompanhar satisfatoriamente a dinâmica do parâmetro κ_c (Figura 4.41f). O novo *setpoint* causou diminuição da conversão de monômero (Figura 4.41g). Em contrapartida, houve aumento da produção de polímero (Figura 4.41h). Esta alteração da dinâmica do processo foi causada pela mudança dos regimes de alimentação. Isso ilustra a dificuldade em prever o comportamento de sistemas não lineares sem um modelo matemático.

No tempo de 4 h o controlador volta para o modo SISO e a falha é efetivamente aplicada na planta. As trajetórias preditas para as variáveis controladas (diamantes cinzas) foram precisas. Em alguns gráficos não é possível enxergá-las, pois elas estão exatamente embaixo da trajetória da planta.

Na Figura 4.42 os valores das variáveis manipuladas durante a avaliação de segurança são mostrados nas linhas tracejadas cinzas partindo de 1 h. Como a simulação de falhas leva em consideração a operação normal da planta, a única variável manipulada é a temperatura da camisa. Identificado um estado de risco, o controlador leva a planta ao novo *setpoint* que, em malha aberta, é representado pelas linhas contínuas cinzas partindo de 4 h. A diferença estacionária entre a resposta do controlador e as trajetórias preditas em malha aberta é uma consequência normal da mudança no comportamento dinâmico do sistema imposta pelo fechamento da malha de controle.

Note que, após a falha, a temperatura do reator se mantém em torno do limite de segurança estabelecido (Figura 4.41a). Para este caso, portanto, o sistema inteligente de segurança foi capaz de reverter uma falha que elevaria a temperatura do reator a cerca 392 K, bem acima do limite de segurança, e que ativaria o *shutdown* de emergência e, consequentemente, causaria um enorme prejuízo econômico. O sistema passou a

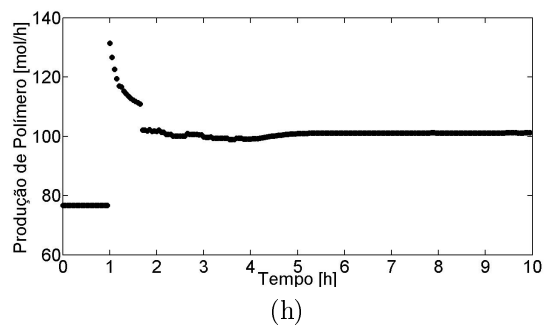
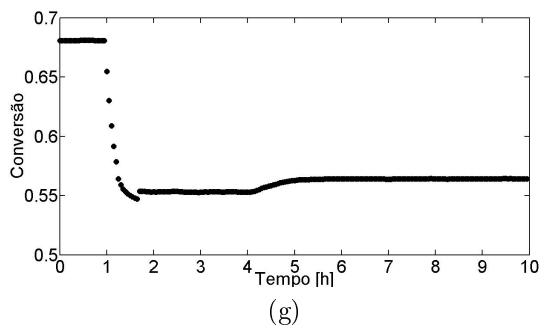
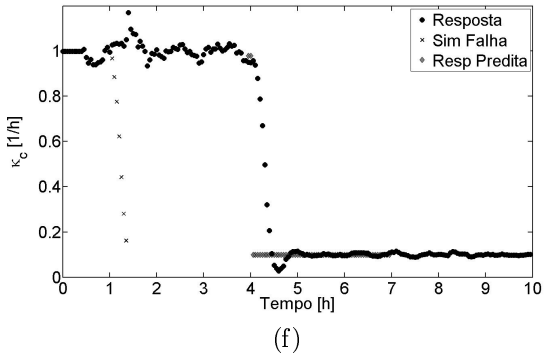
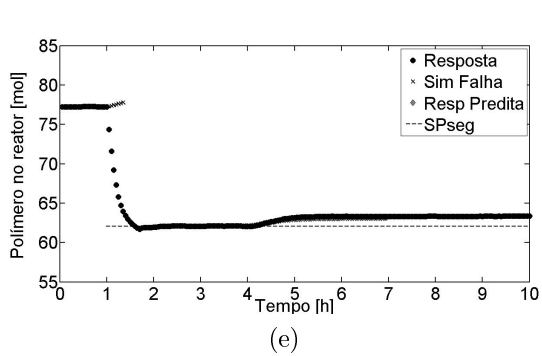
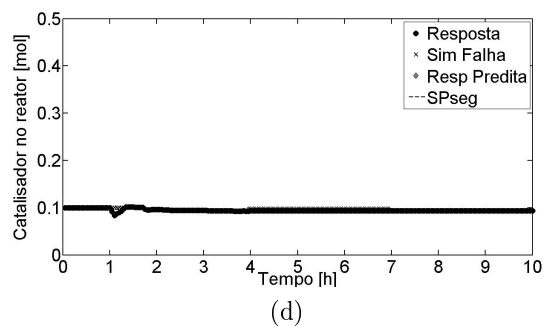
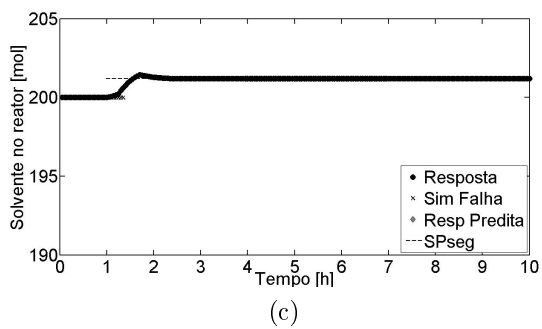
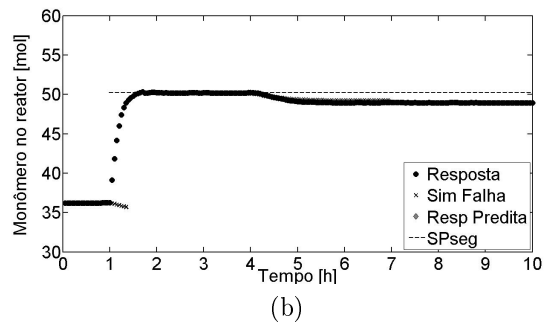
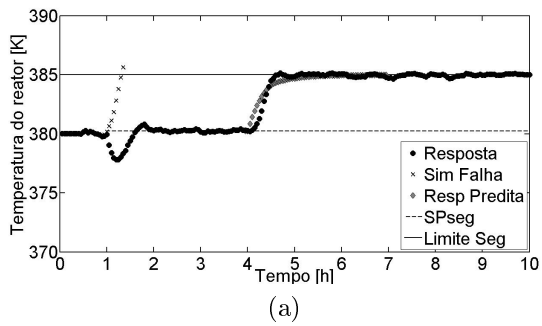


Figura 4.41: Desempenho do sistema integrado de segurança: $\kappa_c = 0,1$. Variáveis Controladas e afins.

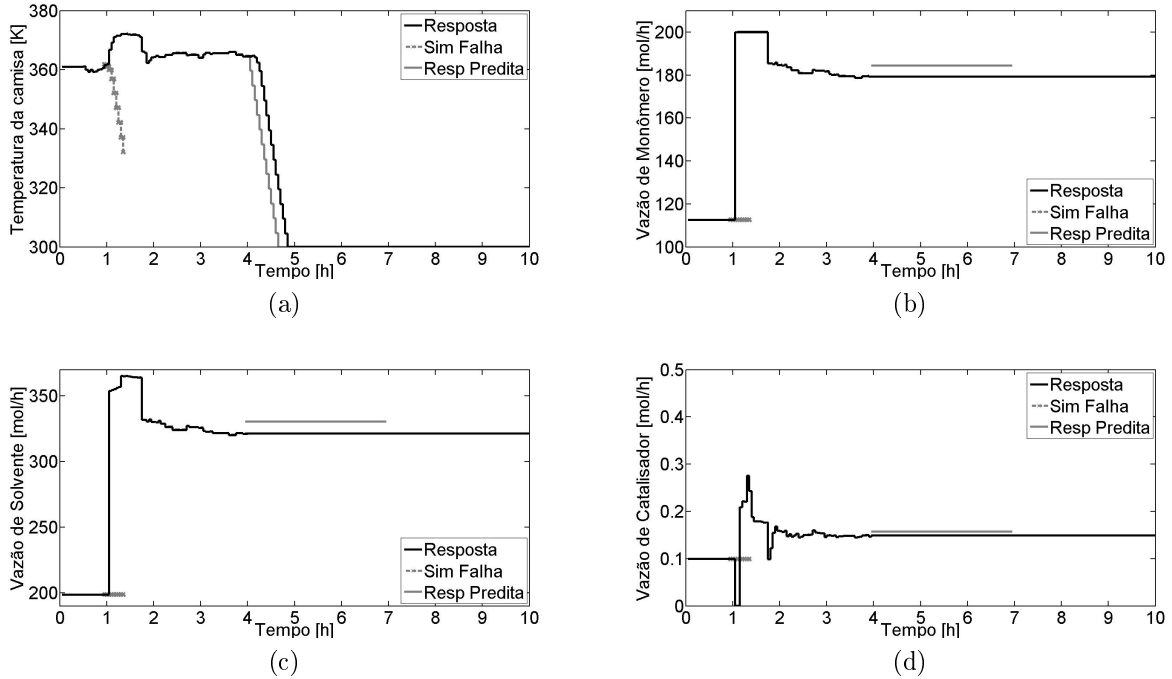


Figura 4.42: Desempenho do sistema integrado de segurança: $\kappa_c = 0,1$. Variáveis Manipuladas.

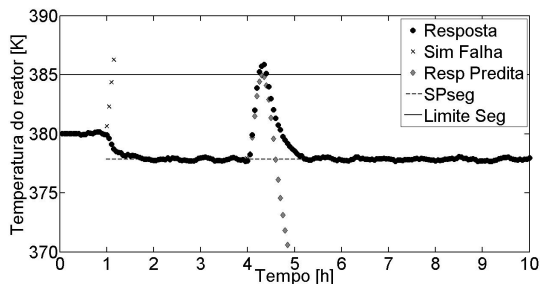
operar com uma conversão mais baixa, mas com aumento de produção, mostrando que as perdas causadas pela ação preventiva de segurança são relativas.

4.5.2 Adição de excesso de catalisador

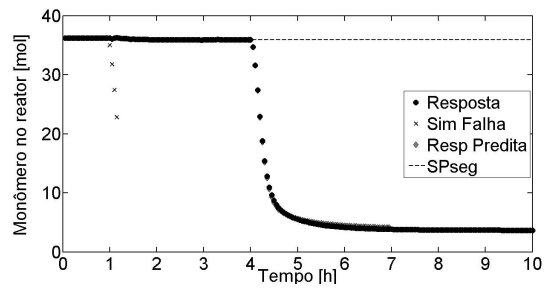
Nesta seção, o desempenho do sistema integrado de segurança é testado para o evento indesejado discutido na Seção 4.4.2, que evidencia os impactos provocados por variações na alimentação de catalisador F_{in}^{Cat} . O evento selecionado para avaliação do sistema de segurança foi o aumento de grau 0,1 para 1,5, cuja dinâmica é mostrada na Figura 4.39 em círculos pretos. Os resultados são mostrados nas Figuras 4.43 e 4.44.

Novamente o sistema de avaliação de risco identifica um risco potencial e ativa o buscador de estados seguros, cuja solução é o novo *setpoint* do controlador. A única mudança perceptível nas variáveis controladas foi uma redução de 2 K na temperatura do reator (Figura 4.43a). A dinâmica da constante de troca térmica foi, novamente, bem acompanhada pelo reconciliador (Figura 4.43f) e, desta vez, o *setpoint* de segurança não provocou alterações na conversão nem na produção de polímero, como pode ser observado nas Figuras 4.43g e 4.43h, respectivamente. Por causa da flutuação natural das estimações do reconciliador, o valor do parâmetro κ_c utilizado na predição (diamantes cinzas) foi a média dos últimos 5 pontos estimados.

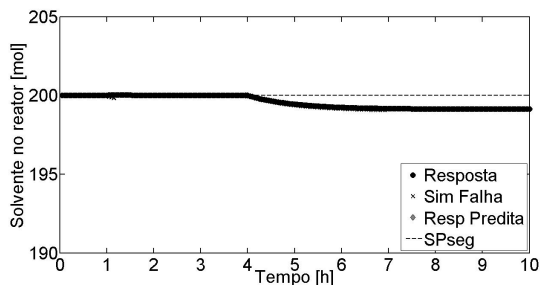
As trajetórias previstas para as falhas foram exatas para quase todas as variáveis



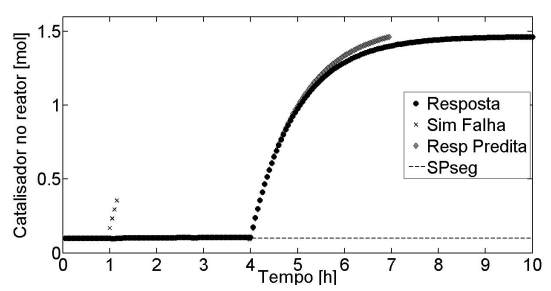
(a)



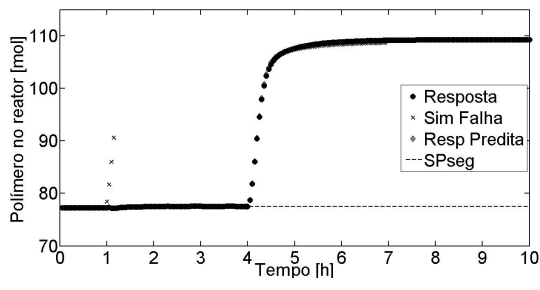
(b)



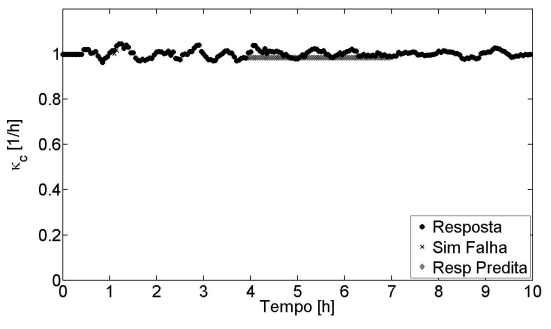
(c)



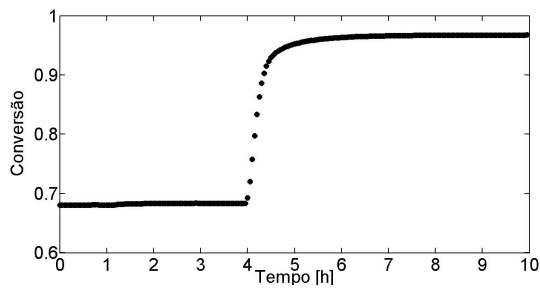
(d)



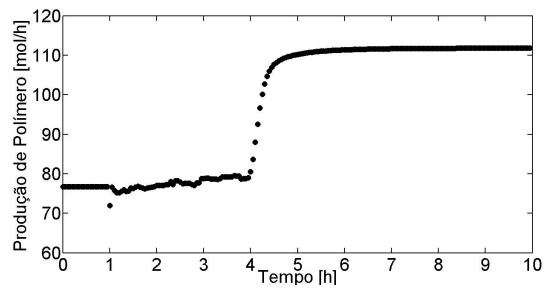
(e)



(f)



(g)



(h)

Figura 4.43: Desempenho do sistema integrado de segurança: $F_{in}^{Cat} = 1,5$. Variáveis Controladas e afins.

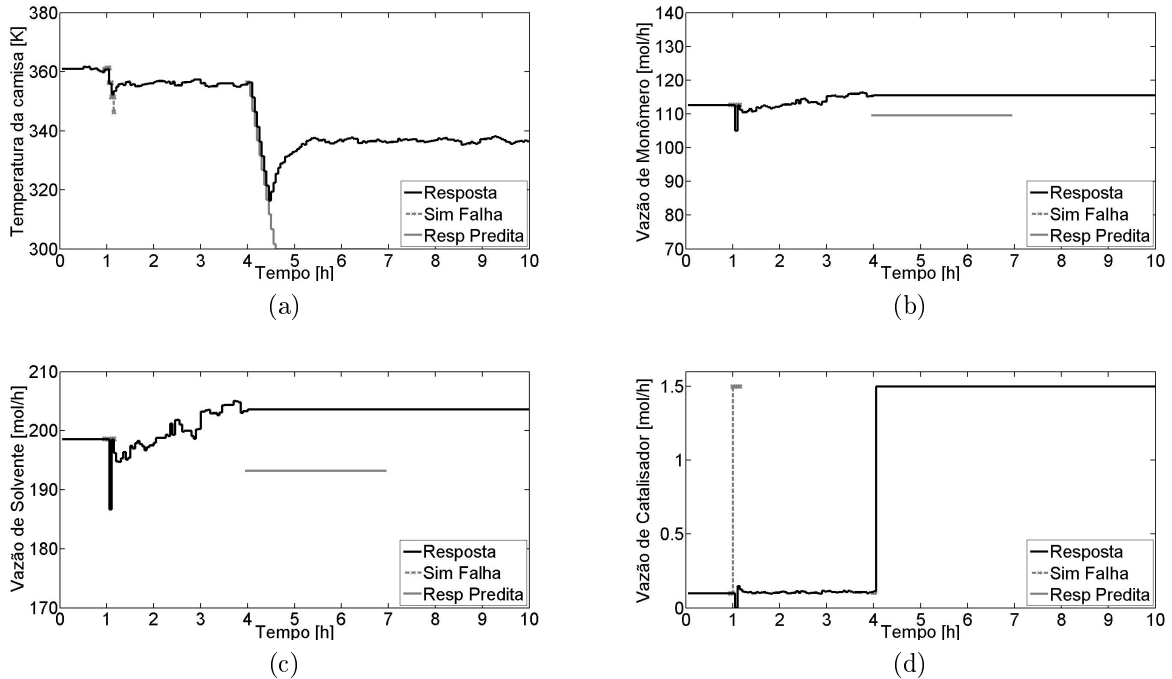


Figura 4.44: Desempenho do sistema integrado de segurança: $F_{in}^{Cat} = 1,5$. Variáveis Manipuladas.

controladas. A única exceção foi para a quantidade de catalisador no reator, que teve uma pequena defasagem em relação à resposta da planta (Figura 4.43d), mas não comprometeu o desempenho do sistema. Tal desvio, provavelmente, se deu em virtude da diferença, mais acentuada que no caso anterior, entre os valores estacionários preditos em malha aberta e a resposta do controlador (Figuras 4.44b e 4.44c).

Note na Figura 4.43a que houve uma pequena violação do limite de segurança. Violações desta magnitude são aceitáveis e esperadas, uma vez que o sistema está sujeito a erros de medição e modelagem que, certamente, influenciam nos cálculos do controlador, do reconciliador e do sistema de segurança. Portanto, o sistema inteligente foi capaz de reverter uma falha, que em condições normais de operação, elevaria a temperatura do reator a quase 390 K.

4.5.3 Perda de controle da temperatura da camisa

Nesta seção, o desempenho do sistema integrado de segurança é testado para o evento indesejado discutido na Seção 4.4.3, que evidencia os impactos provocados por problemas na manutenção da temperatura de camisa T_c . O evento selecionado para avaliação do sistema de segurança foi o travamento da temperatura da camisa em 375 K, cuja dinâmica é mostrada na Figura 4.40 em círculos pretos. Os resultados são mostrados nas Figuras 4.45 e 4.46.

Mais uma vez, o sistema de avaliação de risco identifica um risco potencial e ativa o buscador de estados seguros, cuja solução é o novo *setpoint* do controlador. Neste caso, foram observadas mudanças significativas nas quantidades de monômero e polímero no reator, que tiveram, respectivamente, aumento de 38% e redução de 18% (Figuras 4.46b e 4.46e). Houve diminuição de 3 K na temperatura do reator (Figura 4.46a). A estimação do κ_c foi realizada, novamente, com a média dos últimos 5 pontos e apresentou uma dispersão acentuada a partir de 4 h (Figura 4.46f), mas não comprometeu o desempenho do sistema de segurança. Assim como no primeiro caso, o estado seguro possui menor conversão e maior produtividade (Figuras 4.46g e 4.46h). Destaque para a precisão das trajetórias previstas das variáveis controladas no caso de falha (Figuras 4.46a - 4.46e) e para a diferença bem pequena entre as previsões em malha aberta e a resposta do controlador (Figura 4.45)

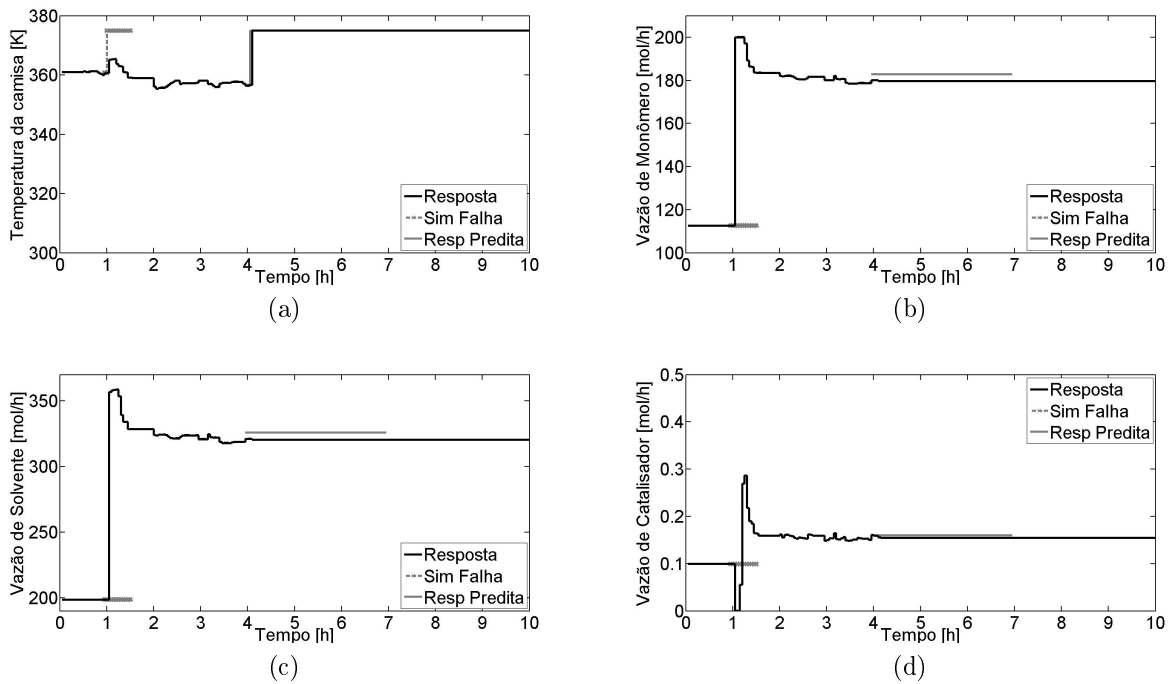
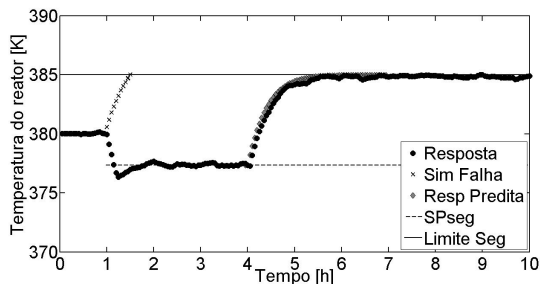


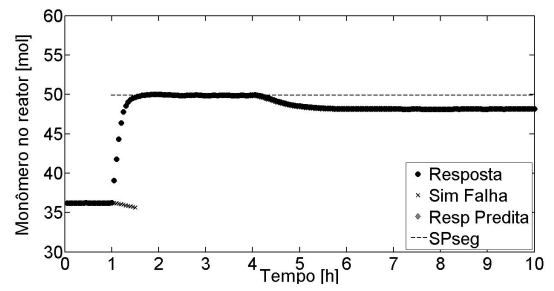
Figura 4.45: Desempenho do sistema integrado de segurança: $T_c = 375$. Variáveis Manipuladas.

Novamente, o sistema inteligente de segurança foi capaz de reverter uma falha que causaria o *shutdown* do reator. A temperatura máxima atingida a partir do estado seguro estabiliza em torno do limite de segurança (Figura 4.46a). Em condições normais, a temperatura do reator atingiria 388 K.

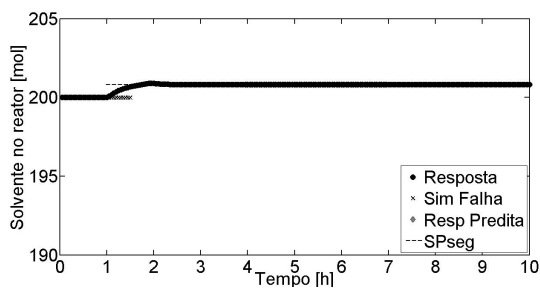
O tempo médio gasto pelo sistema de segurança para simular as falhas foi de 2 minutos. Já o tempo para encontrar o estado seguro foi, em média, menor que 1 minuto. Este resultado mostra que a metodologia proposta pode ter aplicação em tempo real.



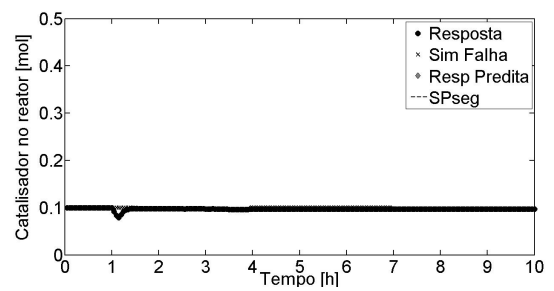
(a)



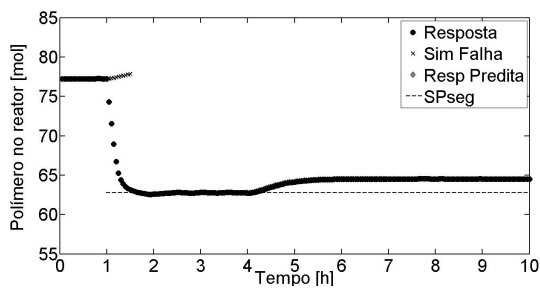
(b)



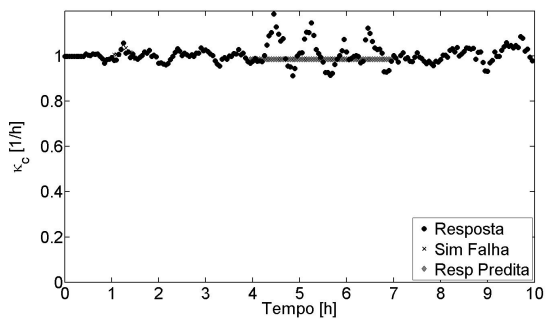
(c)



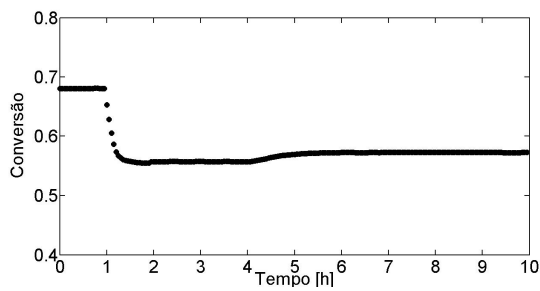
(d)



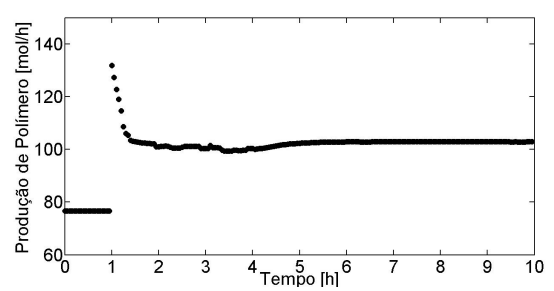
(e)



(f)



(g)



(h)

Figura 4.46: Desempenho do sistema integrado de segurança: $T_c = 375$. Variáveis Controladas e afins.

4.6 Robustez frente a incertezas

Todas as simulações anteriores foram realizadas partindo do princípio de que o processo possui um bom sistema de detecção de falhas. Em outras palavras, o processo é capaz de identificar a origem e a magnitude das falhas no instante em que elas acontecem. Nas simulações realizadas neste trabalho, o reconciliador foi informado sobre as falhas, deixando o parâmetro κ_c e os estados do modelo como as únicas incógnitas. Mas, e se o sistema de detecção não for eficaz? Afinal, detectar falhas não é uma tarefa simples. Portanto, nesta seção, o desempenho do sistema integrado de segurança foi testado sob incertezas nas medidas da vazão de alimentação de catalisador e da temperatura do reator.

4.6.1 Incerteza sobre a vazão de catalisador

A incerteza considerada nesta seção equivale a uma falha no medidor da vazão de alimentação de catalisador. Assim, no instante 4 h, além do travamento da válvula que controla a alimentação de catalisador (mesma situação simulada na Seção 4.5.2), ocorre também um problema na medição da vazão da substância, que passa a transmitir sempre o mesmo valor. Numa planta industrial, tal combinação impediria a ativação do alarme de alta vazão de catalisador, deixando os operadores sem condições de decidir o que fazer.

Nesta simulação, o reconciliador não foi alimentado com os valores verdadeiros da vazão de catalisador, mas com o último valor antes da efetivação da falha. Os resultados são mostrados nas Figuras 4.47 e 4.48, em que os valores estimados pelo reconciliador – círculos pretos – são comparados aos valores reais da planta (livres de incertezas) – quadrados brancos.

Nas Figuras 4.48a - 4.48f as trajetórias estimadas pelo reconciliador são muito próximas às trajetórias reais da planta até o instante 4 h. A partir deste momento, a falha no medidor é inserida e o reconciliador tenta compensá-la através do parâmetro κ_c que, apesar de ter seu valor real constante e igual a 1, exibe um perfil estimado oscilatório em torno de 0,5. Mesmo com esta compensação, todos os estados, com exceção da temperatura, são estimados como constantes, porém, seus valores reais mudam. A temperatura do reator foi o único estado estimado com precisão por causa da própria estrutura do algoritmo de reconciliação, que tenta aproximar $T_{estimado}$ do T_{medido} em função do κ_c . Dentre todos os estados, somente T é função de κ_c (Equação 3.1), portanto, se não houver falha na medição do T_{medido} , $T_{estimado}$ sempre estará próximo à temperatura real do reator, mesmo que o reconciliador estime valores irreais para κ_c .

Por ser alimentado pelos resultados do reconciliador, o controlador enxerga somente a suposta variação do κ_c , calculando um perfil igualmente oscilatório para

a temperatura da camisa. Comparando as Figuras 4.48a, 4.48f e 4.47a, observa-se que às 4 h a temperatura do reator começa a subir. O reconciliador não enxerga o aumento de F_{in}^{Cat} , então, tenta compensar diminuindo o κ_c . Com o desvio de T ao *setpoint*, o controlador baixa a temperatura da camisa na tentativa de trazer a variável controlada de volta ao valor desejado. Note que, em torno de 4,5 h, o κ_c estimado bate na restrição inferior e, no mesmo instante, o controlador mantém T_c constante, uma vez que, valores nulos de κ_c anulam também os efeitos de T_c no modelo do controlador. Quando κ_c volta a um valor não nulo, o controlador continua reduzindo T_c até que T esteja próximo ao *setpoint*. A partir de então, κ_c exhibe comportamento oscilatório, provocando também um comportamento oscilatório em T_c , porém, inversamente proporcional a κ_c .

Os cálculos da conversão e da produção de polímero foram realizados com os dados reais do processo. A Figura 4.47d ilustra a vazão real de catalisador alimentada ao reator (linha tracejada preta) – que passa sobre a linha da resposta predita para o caso de falha, como esperado – e a vazão transmitida ao reconciliador (linha contínua preta).

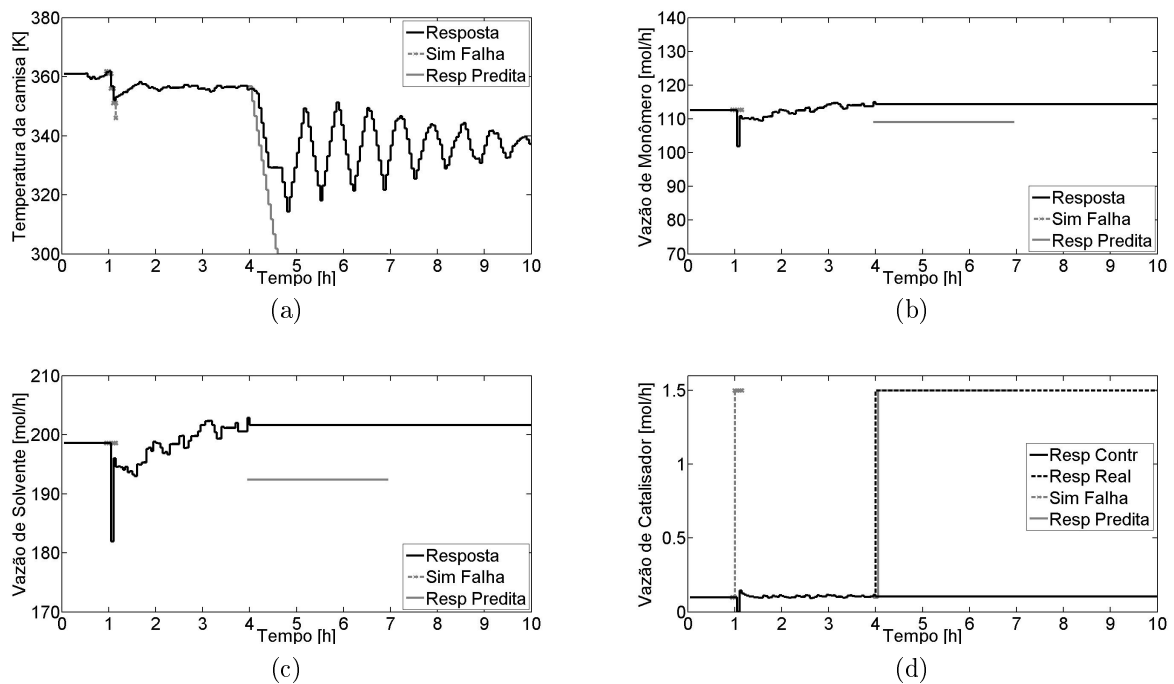


Figura 4.47: Robustez do sistema integrado de segurança: falha no medidor de vazão de catalisador. Variáveis Manipuladas.

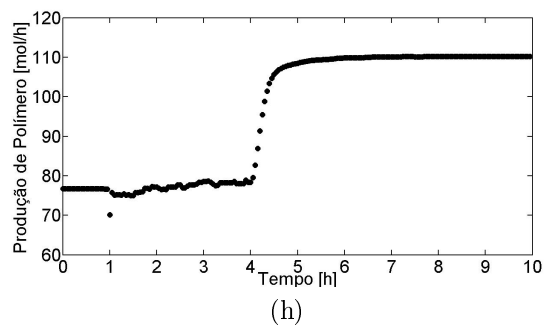
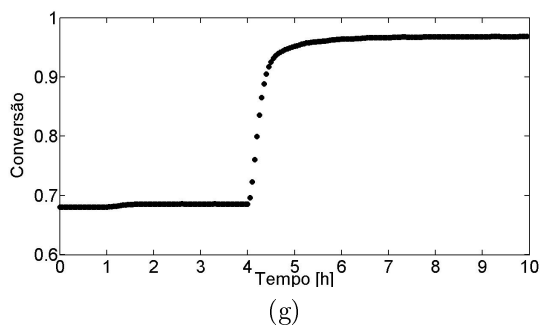
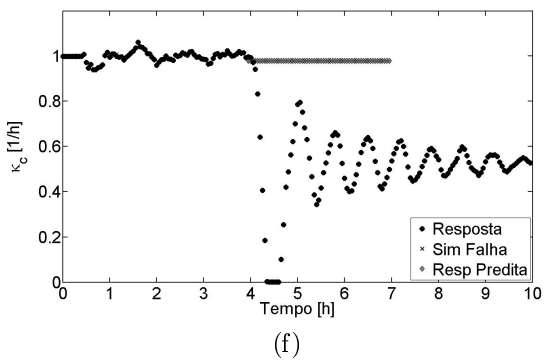
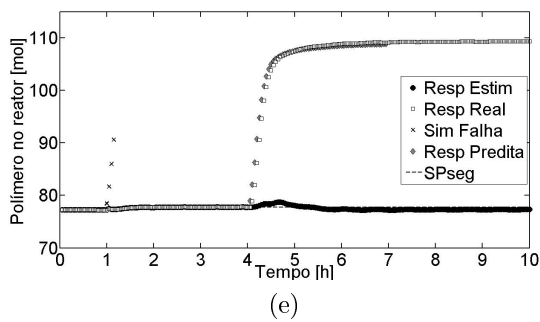
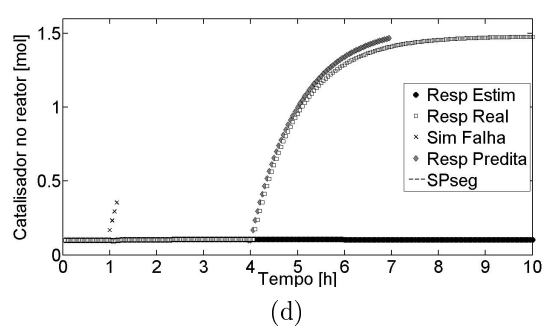
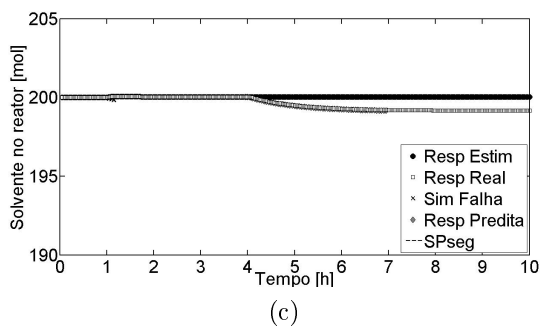
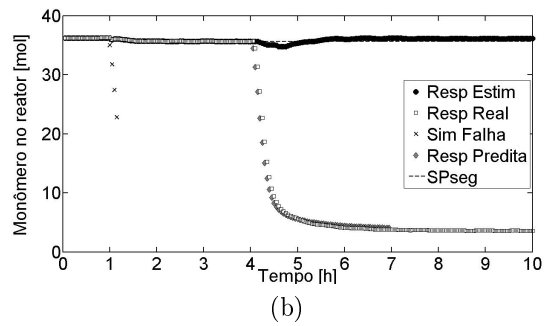
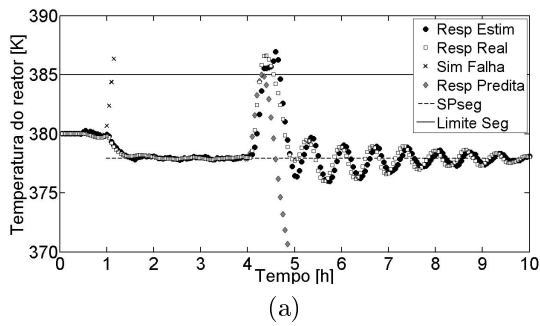


Figura 4.48: Robustez do sistema integrado de segurança: falha no medidor de vazão de catalisador. Variáveis Controladas e afins.

Numa planta industrial, os operadores teriam acesso somente aos estados estimados pelo reconciliador, visto que a única variável medida é a temperatura, e neste caso, não perceberiam todas as alterações na dinâmica do reator, apenas as variações na constante de troca térmica. Essa visão limitada acerca do processo poderia levá-los a tomar decisões erradas que, por sua vez, poderiam causar sérios problemas. Portanto, apesar da falha de medição da vazão de catalisador, o estado seguro foi capaz de manter o processo dentro dos limites de segurança, ratificando a importância de sistemas de segurança mais inteligentes, baseados em modelos de processo, para uma operação segura.

4.6.2 Incerteza sobre a temperatura do reator

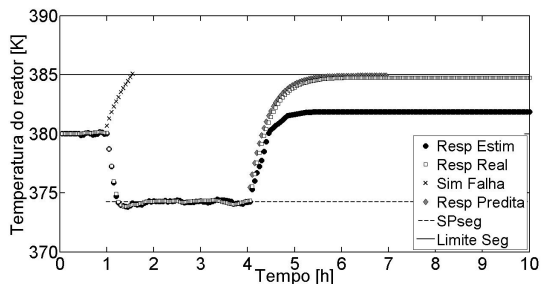
A incerteza considerada nesta seção equivale a uma falha no medidor da temperatura do reator. Assim, no instante 4 h, além do travamento da temperatura da camisa (mesma situação simulada na Seção 4.5.3), ocorre também um problema na medição da temperatura do reator, que passa a transmitir sempre o mesmo valor. Numa planta industrial, tal combinação poderia impedir a ativação do alarme de temperatura alta, deixando os operadores sem condições de decidir o que fazer.

Nesta simulação, o reconciliador não foi alimentado com os valores medidos na planta ($T_{real} + \text{ruído}$) em cada instante de amostragem, mas com o último valor antes da efetivação da falha ($T_{real}(4h) + \text{ruído}$). A partir de então, o valor base $T_{real}(4h)$ foi mantido, mas o ruído continuou variando em cada amostragem. Os resultados são mostrados nas Figuras 4.49 e 4.50, onde os valores estimados pelo reconciliador – círculos pretos – são comparados aos valores reais da planta (livres de incertezas) – quadrados brancos.

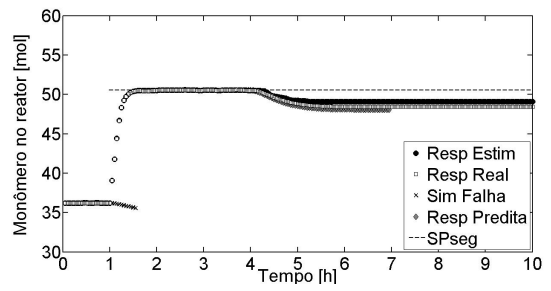
Observa-se que o reconciliador, mesmo sendo alimentado continuamente com uma temperatura em torno dos 374 K foi capaz de estimar razoavelmente bem todos os estados. A única exceção foi a temperatura do reator que, por causa do limite imposto ao parâmetro κ_c , não pode ser corretamente estimada. Neste caso, κ_c novamente possui valor real constante e igual a 1. Destaque para a temperatura real do reator (Figura 4.49a), que não ultrapassa o limite de segurança.

4.6.3 Considerações finais

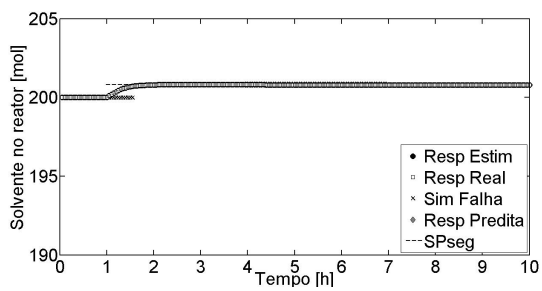
Neste capítulo foram discutidos os resultados da aplicação do sistema de segurança proposto para diversos casos. Mas antes disso, o passo a passo de toda a sintonia do controlador e do reconciliador foram mostradas em detalhes, deixando claro que ambos os algoritmos funcionam bem conforme as exigências e que, apesar de simples, os procedimentos de sintonia se mostraram plenamente satisfatórios. Foram discutidos também alguns comportamentos inseguros comuns em processos



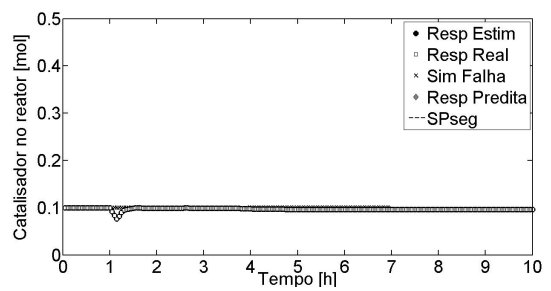
(a)



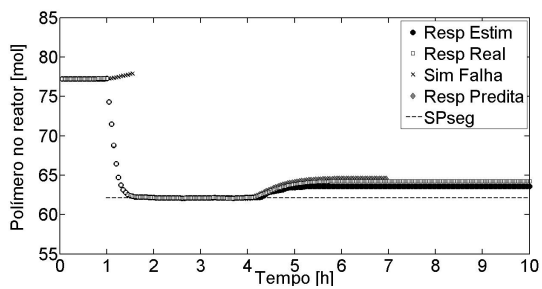
(b)



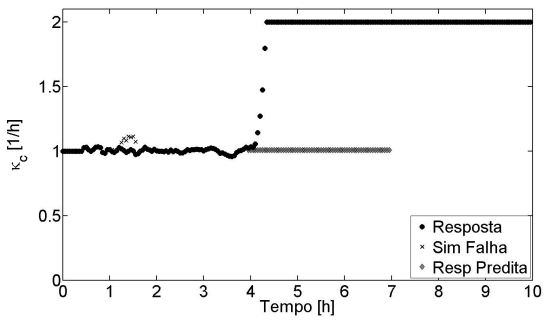
(c)



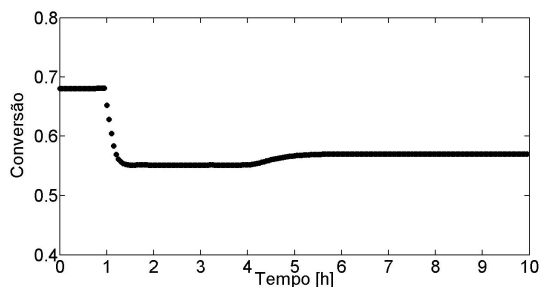
(d)



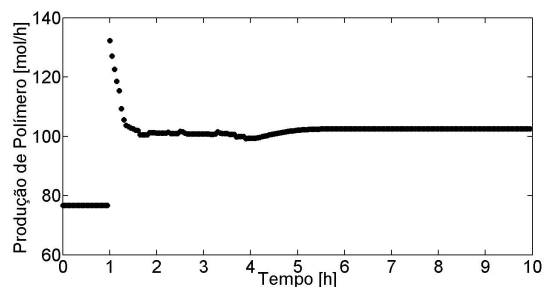
(e)



(f)



(g)



(h)

Figura 4.49: Robustez do sistema integrado de segurança: falha no medidor de temperatura do reator. Variáveis Controladas e afins.

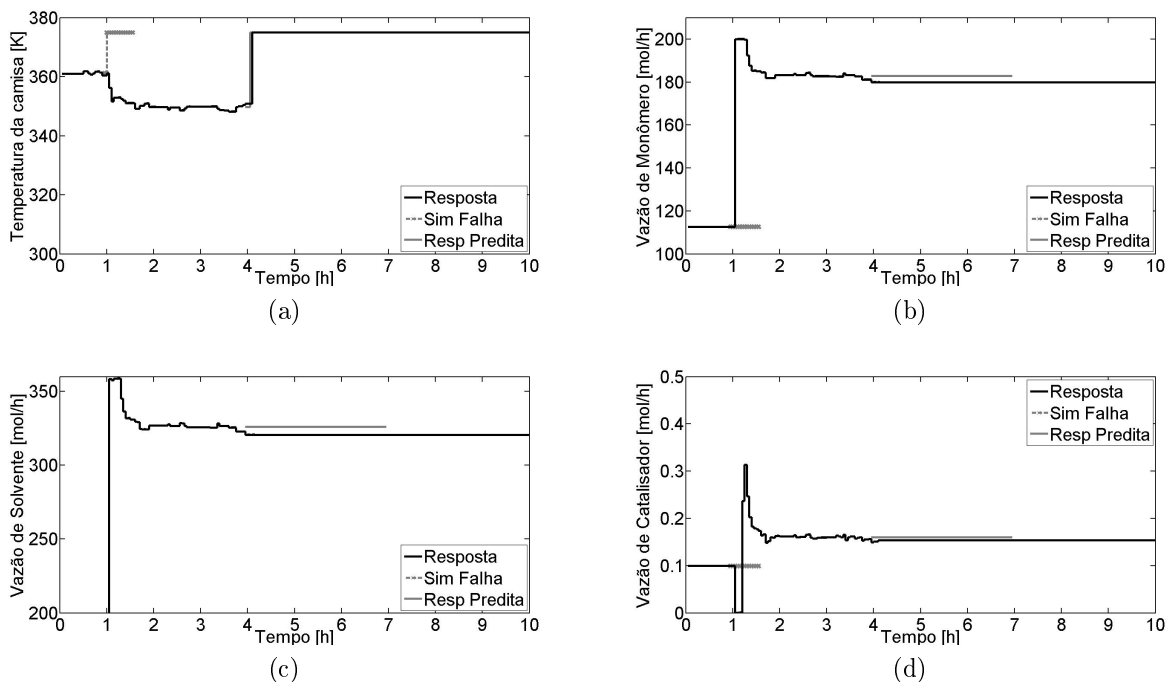


Figura 4.50: Robustez do sistema integrado de segurança: falha no medidor de temperatura do reator. Variáveis Manipuladas.

químicos, a partir dos quais, o desempenho do sistema inteligente de segurança foi testado. Estas simulações foram apenas ilustrações do comportamento do algoritmo proposto. Os valores das variáveis não tem compromisso com casos reais, mas o comportamento qualitativo das mesmas é importante.

O esquema de segurança foi capaz de manter o controle do processo em todos os casos, inclusive em situações onde há propagação de falhas, tornando possível evitar *shutdowns* desnecessários. A decisão mais difícil, contudo, se dá no momento em que uma situação de risco potencial é identificada. Não há como prever se a falha realmente ocorrerá, mesmo com métodos estatísticos para cálculo de probabilidades de falha. Portanto, atuar preventivamente sobre o sistema não é uma decisão simples. A proposta deste trabalho é atuar em situações envolvendo variáveis críticas, cuja severidade e frequência tornem o cenário de alto risco.

Considerando a capacidade computacional de um computador doméstico Intel[®] Core[™] i3-2330M 2,2GHz 4GB RAM, o algoritmo proposto neste trabalho efetuou os cálculos de análise de risco e cálculo do estado seguro em menos de quatro minutos para todos os casos, característica que torna o método proposto aplicável em tempo real.

Capítulo 5

Conclusões

O desenvolvimento dos sistemas de segurança na indústria foi historicamente motivado por acidentes trágicos. Os sistemas atuais são baseados em normas rigorosas que se estendem desde a operação e projeto de equipamentos até o comportamento diário dos profissionais da indústria. Entretanto, apesar da grande complexidade dos processos químicos, o uso de modelos matemáticos não lineares ainda é muito escasso. Por isso, os sistemas de segurança geralmente são conservadores e ainda muito dependentes de decisões humanas, tornando-os vulneráveis a comportamentos dinâmicos não lineares. Por causa desta fragilidade, o uso do *shutdown* de emergência torna-se comum para situações onde as falhas não são detectadas e limites estreitos de segurança são violados, gerando grandes prejuízos econômicos.

Este trabalho traz uma alternativa ao *shutdown* de emergência na tentativa de evitar desligamentos desnecessários e aumentar o tempo de operação anual da planta. Caso o algoritmo proposto não encontre uma solução viável, o *shutdown* torna-se realmente necessário. E os resultados mostraram que o algoritmo proposto pode ser viável se houver um modelo confiável do processo.

O NMPC é uma estratégia de controle fundamental para o sucesso do sistema inteligente de segurança. Através dele é possível prever o comportamento dinâmico do processo e antecipar ações de controle. Mas para seu bom funcionamento a sintonia deve ser adequada. O método de sintonia baseado na influência de cada variável na função objetivo do controlador foi simples e obteve desempenho satisfatório. Os horizontes de controle e predição não produziram efeitos significativos no desempenho do controlador. Já a estratégia de atualização do modelo de predição mostrou-se eficaz para adaptação do controlador às variações da constante de troca térmica.

A atualização do modelo do processo foi possível graças à precisão do esquema de reconciliação de dados que, apenas com medidas de temperatura, foi capaz de acompanhar a dinâmica do sistema em situações adversas. A sintonia do horizonte de reconciliação baseada no desempenho frente ao ruído foi simples e eficaz.

O sistema inteligente de segurança se mostrou eficaz para diversas falhas que

poderiam acarretar em problemas de troca térmica, problemas na camisa do reator e adição excessiva de catalisador. Todos esses eventos são indesejados porque podem provocar um *runaway* e a conseqüente perda de controle. Em todos os cenários discutidos na Seção 4.5, o esquema de segurança obteve sucesso sem grandes mudanças no *setpoint* das variáveis controladas, contudo, o perfil de alimentação do reator sofreu grandes mudanças para os casos de perda da capacidade de refrigeração e perda de controle da temperatura da camisa. No caso da adição de excesso de catalisador houve uma pequena violação do limite de segurança devido ao ruído do processo, todavia, pequenos desvios sempre são esperados em plantas industriais, uma vez que, além do ruído, existem incertezas de modelagem e perturbações não controladas.

As simulações de propagação de falhas mostraram que em alguns casos o sistema inteligente de segurança pode garantir a segurança mesmo que ocorram falhas nos medidores, somadas aos comportamentos inseguros. Nestes casos, os operadores não teriam ciência do estado real do sistema e, portanto, poderiam tomar decisões precipitadas. Uma vez que o processo esteja em um estado seguro, mesmo com essas problemas, os limites de segurança não são violados.

Para finalizar, o tempo de simulação gasto nas avaliações de risco e nos cálculos do estado seguro foi sempre inferior à quatro minutos, considerando a capacidade computacional de um computador doméstico Intel[®] Core[™] i3-2330M 2,2GHz 4GB RAM, o que torna o método proposto aplicável em tempo real.

5.1 Sugestões para trabalhos futuros

Para trabalhos futuros, sugere-se que o algoritmo de segurança seja testado para modelos de processo validados ou *benchmarks*, analisando o impacto do sistema de segurança em parâmetros de qualidade do processo.

Outro ponto importante é a simulação de diversos cenários de falhas simultaneamente, através de uma implementação para processamento paralelo, uma vez que computadores *multicore* são cada vez mais comuns no mercado. Isto torna o algoritmo completo viável para aplicações em tempo real.

Como neste trabalho foi utilizado um algoritmo de otimização de caráter determinístico, há a possibilidade de os estados seguros encontrados serem mínimos locais. Além disso, outras combinações entre as variáveis podem gerar o mesmo resultado. Portanto, um estudo para ocorrência de múltiplas soluções também é importante em trabalhos futuros.

Referências Bibliográficas

ALICI, S.; EDGAR, T. F. Nonlinear dynamic data reconciliation via process simulation software and model identification tools. **Industrial and Engineering Chemistry Research**, v. 41, p. 3984–3992, 2002.

AMPELLI, C.; DI BELLA, D.; MASCHIO, G.; RUSSO, A. Calorimetric study of the inhibition of runaway reactions during methylmethacrylate polymerization processes. **Journal of Loss Prevention in the Process Industries**, v. 19, n. 5, p. 419–424, 2006.

BARADITS, G. **Safety instrumented system management**. 2010. 160 f. Tese (Doutorado) - University of Pannonia, Veszprém, Hungria. 2010. Disponível em: <http://konyvtar.uni-pannon.hu/doktori/2010/Baradits_Gyorgy_dissertation.pdf>. Acesso em: 20 abr. 2014.

CASSON, V.; LISTER, D. G.; MILAZZO, M.F.; MASCHIO, G. Comparison of criteria for prediction of runaway reactions in the sulphuric acid catalyzed esterification of acetic anhydride and methanol. **Journal of Loss Prevention in the Process Industries**, v. 25, n. 1, p. 209–217, 2012.

DAKSHINAMOORTHY, D.; KHOPKAR, A. R.; LOUVAR, J. F.; RANADE, V. V. CFD simulations to study shortstopping runaway reactions in a stirred vessel. **Journal of Loss Prevention in the Process Industries**, v. 17, n. 5, p. 355–364, 2004.

DESHPANDE, A.; PATWARDHAN, S. C.; NARASIMHAN, S. Integrating fault diagnosis with nonlinear model predictive control. In: FINDEISEN, R.; ALLGÖWER, F.; BIEGLER, L. (Eds.), **Assessment and future directions of nonlinear model predictive control**. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007. p. 513–521.

DUIJM, N. J.; GOOSSENS, L. Quantifying the influence of safety management on the reliability of safety barriers. **Journal of Hazardous Materials**, v. 130, p. 284–292, 2006.

ECKERMAN, I. **The Bhopal saga: causes and consequences of the world's largest industrial disaster**. Hyderabad , Índia: Universities Press, 2005. Disponível em: <<http://books.google.se/books?id=rvn7ybZUo4kC&printsec=frontcover#v=onepage&q&f=false>>. Acesso em: 20 abr. 2014.

EEMUA. **Alarm systems: A guide to design, management and procurement**. EE-

MUA Publication No. 191 (2nd ed.), 2007. London: Engineering Equipment and Materials Users' Association.

EUROPEAN COMMISSION. **Chemical Accidents (Seveso I, II and III):** Prevention, Preparedness and Response. Disponível em: <<http://ec.europa.eu/environment/seveso/>>. Acesso em: 24 fev. 2014.

FREDERICKSON, A. A. **The Layer of Protection Analysis (LOPA) method:** Look for best practices and guidelines on how to use the LOPA method as an alternative to mitigate risks. p. 1–9, 2002.

GARCIA, C. E.; PRETT, D. M.; MORARI, M. Model Predictive Control: theory and practice - a survey. **Automatica**, v. 25, n. 3, p. 335–348, 1989.

GRAF, H. & SCHMIDT-TRAUB, H. An integrated approach to early process hazard identification of continuous and batch plants with statechart modelling and simulation. **Computers and Chemical Engineering**, v. 25, p. 61-72, 2001.

GYGAX, R. Chemical reaction engineering for safety. **Chemical Engineering Science**, v. 43, n. 8, p. 1759-1771, 1988.

HAWKINS, R. **What is a Quantitative Risk Assessment?** Disponível em: <<http://safetymatters.aonfp.com/2012/Volume-1/risk.aspx>>. Acesso em: 16 abr. 2014.

HOCHHEIMER, D. A.; LEMKOWITZ, S. M.; GORT, J.; BOERS, M.; PASMAN, H. J. Impact of safety culture on layers of protection. In: 2nd International Conference on Safety & Environment in Process Industry, CISAP-2, May 21–24, 2006, Naples, Italy. **Chemical Engineering Transactions**, v. 9, p. 65–70, 2006.

HOWELL, P. P. Plant Explosion Emphasizes Importance of Implementing PSM. **Process Safety Progress**, v. 29, n. 2, p. 144-149, 2010.

IEC. (1999). International Electrotechnical Commission (IEC). IEC 61508, **Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems**. Geneva, Switzerland.

IEC. (2003). International Electrotechnical Commission (IEC), IEC 61511, **Functional Safety: Safety Instrumented Systems for the Process Sector**. Geneva, Switzerland.

ISA. (1996). International Society of Automation (ISA). ANSI/ISA 84.01-1996, **Application of Safety Instrumented Systems for the Process Industries**. Research Triangle Park, NC.

ISA. (2004). International Society of Automation (ISA). ANSI/ISA 84.00.01-2004, **Functional Safety: Safety Instrumented Systems for the Process Industry Sector**. Research Triangle Park, NC.

ISA. (2009). **Management of alarm systems for the process industries**. Technical Report ANSI/ISA-18.2-2009. 67 Alexander Drive, P.O. Box 12277, Research Triangle Park, North Carolina 27709: International Society of Automation, ISA.

JANG, S.; JOSEPH, B.; MUKAI, H. Comparison of two approaches to on-line parameter and state estimation of nonlinear systems. **Industrial Engineering and Chemical Process Design Development**, v. 25, p. 809-814, 1986.

KLOOS M; PESCHKE J. Consideration of human actions in combination with the probabilistic dynamics method Monte Carlo dynamic event tree. **Journal of Risk and Reliability**, v. 222, p. 303–313, 2008. LABEAU, P. E.; SMIDTS, C.; SWAMINATHAN, S. Dynamic reliability: towards an integrated platform for probabilistic risk assessment. **Reliability Engineering & System Safety**, v. 68, n. 3, p. 219–254, 2000.

LABOVSKÝ, J.; ŠVANDOVÁ, Z.; MARKOŠ, J.; JELEMENSKÝ, L. Mathematical model of a chemical reactor: useful tool for its safety analysis and design. **Chemical Engineering Science**, v. 62, n. 18-20, p. 4915–4919, 2007a.

LABOVSKÝ, J.; ŠVANDOVÁ, Z.; MARKOŠ, J.; JELEMENSKÝ, L. Model-based HAZOP study of a real MTBE plant. **Journal of Loss Prevention in the Process Industries**, v. 20, n. 3, p. 230–237, 2007b.

LAURENT, J. L. **Explosion at the AZF factory in Toulouse**: the general inspectorate for the environment hands over its report to Yves Cochet. Press Release. Paris. 52 p. 2001. Disponível em: <<http://www.hse.gov.uk/landuseplanning/toulouse.pdf>>. Acesso em: 16 abr. 2014.

LAWLEY, H. G. Operability studies and hazard analysis. **Chemical Engineering Progress**, v. 70, p. 105–116, 1974.

LIEBMAN, M.J.; EDGAR, T.F.; LASDON, L.S. Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. **Computers & Chemical Engineering**, v. 16, p. 963-986, 1992.

LUYBEN, W. L. Use of dynamic simulation for reactor safety analysis. **Computers & Chemical Engineering**, v. 40, p. 97–109, 2012.

MANDERS, L. G.; MEISTER, M.; BAUSA, J.; HUNGENBERG, K.-D. Online model-based process safety concepts in polymerization. **Macromolecular Symposia**, v. 302, n. 1, p. 289–296, 2011.

MANENTI, F. Considerations on nonlinear model predictive control techniques. **Computers & Chemical Engineering**, v. 35, p. 2491–2509, 2011.

MARHAVILAS, P. K.; KOULOURIOTIS, D.; GEMENI, V. Risk analysis and assessment methodologies in the work sites: On a review, classification and comparative study of the scientific literature of the period 2000–2009. **Journal of Loss Prevention in the Process Industries**, v. 24, n. 5, p. 477–523, 2011.

MARSEGUERRA, M.; ZIO, E. Monte Carlo approach to PSA for dynamic process systems. **Reliability Engineering & System Safety**. v. 52, n. 3, p. 227–241, 1996.

MATLAB, 2011, version 7.13.0 (R2011b). Natick, Massachusetts, The MathWorks Inc.

- MAURYA, M. R; RENGASWAMY, R; VENKATASUBRAMANIAN, V. A signed directed graph and qualitative trend analysis-based framework for incipient fault diagnosis. **Chemical Engineering Research and Design**, v. 85, n. A10, p. 1407–1422, 2007.
- MEADOWS, E. S.; RAWLINGS, J. B. Model predictive control. In: Henson, M. A.; Seborg, D. E. (Eds), **Nonlinear process control**. Englewood Cliffs, NJ: Prentice-Hall, 1997. Capítulo 5, p. 233—310.
- MEEL, A.; SEIDER, W. D. Real-time risk analysis of safety systems. **Computers & Chemical Engineering**, v. 32, n. 4-5, p. 827–840, 2008.
- MERGEL, M. **Baia Mare Cyanide Spill**. Toxipedia. 2011. Disponível em: <<http://www.toxipedia.org/display/toxipedia/Baia+Mare+Cyanide+Spill>>. Acesso em: 25 fev. 2014.
- MOLNÁR, A.; MARKOŠ, J.; JELEMENSKÝ, L. Some considerations for safety analysis of chemical reactors. **Chemical Engineering Research and Design**, v. 83, n. 2, p. 167–176, 2005.
- PASMAN, H. J.; JUNG, s.; PREM, K; ROGERS, W.J.; YANG, X. Is risk analysis a useful tool for improving process safety? **Journal of Loss Prevention in the Process Industries**, v. 22, p. 769–777, 2009.
- PODOFILLINI, L.; DANG, V. N. Conventional and dynamic safety analysis: Comparison on a chemical batch reactor. **Reliability Engineering & System Safety**, v. 106, p. 146–159, 2012.
- PRATA, D.M.; SCHWAAB, M.; LIMA, E.L.; PINTO, J.C. Nonlinear dynamic data reconciliation and parameter estimation through particle swarm optimization: Application for an industrial polypropylene reactor. **Chemical Engineering Science**, v. 64, n. 18, p. 3953-3967, 2009.
- REIS, L. L. G. Controle tolerante com reconfiguração estrutural acoplado a sistema de diagnóstico de falha. 2008. 199 f. Dissertação (mestrado) - Universidade Federal de Uberlândia, Uberlândia, Brasil, 2008.
- SAGAN, S. D. **The limits of safety: organizations, accidents, and nuclear weapons**. Princeton University Press. New Jersey. 1993.
- SECCHI, A. R. **Differential-Algebraic System Solver in C**. 1992–2007. Disponível em: <<http://www.enq.ufrgs.br/enqlib/numeric/>>. Acesso em: 29 abr. 2014.
- SOUZA, P. M.; SOARES, M.; AMARAL, M. M.; LIMA, E. L.; PINTO, J. C. Data reconciliation and control in styrene-butadiene emulsion polymerizations. **Macromolecular Symposia**, v. 302, p. 80–89, 2011.
- SRINIVASAN, R.; VENKATASUBRAMANIAN, V. Petri net-Digraph models for automating HAZOP analysis of batch process plants. **Computers & Chemical Engineering**, v. 20, Supplement 1, p. 719–725, 1996.

STROZZI, F.; ZALDÍVAR, J. M. A general method for assessing the thermal stability of batch chemical reactors by sensitivity calculation based on Lyapunov exponents. **Chemical Engineering Science**, v. 49, n. 94, p. 2681–2688, 1994.

SUMMERS, Angela E. **Differences between IEC 61511 and ISA 84.01-1996**. The Instrumentation, Systems and Automation Society (ISA). ISA Expo 2003, Houston, Texas, 2003.

SWISSINFO. **Police moot new probe into Schweizerhalle blaze**. Disponível em: <http://www.swissinfo.ch/eng/Police_moot_new_probe_into_Schweizerhalle_b blaze.html?cid=1762686>. Acesso em: 25 fev. 2014.

VAIDHYANATHAN, R.; VENKATASUBRAMANIAN, V. Digraph-based models for automated HAZOP analysis. **Reliability Engineering & System Safety**, v. 50, n. 1, p. 33–49, 1995.

VENKATASUBRAMANIAN, V.; RENGASWAMY, R.; YIN, K. A review of process fault detection and diagnosis Part I : Quantitative model-based methods. **Computers & Chemical Engineering**, v. 27, p. 293–311, 2003.

VENKATASUBRAMANIAN, V.; ZHAO, J. S.; VISWANATHAN, S. Intelligent systems for HAZOP analysis of complex process plants. **Computers & Chemical Engineering**, v. 24, n. 9–10, p. 2291–2302, 2000.

VIEIRA, R. A. M.; EMBIRUÇU, M; SAYER, C.; PINTO, J. C.; LIMA, E. L. Control strategies for complex chemical processes: applications in polymerization processes. **Computers & Chemical Engineering**, v. 27, p. 1307-1327, 2003.

VOOGD, H. Disaster Prevention in Urban Environments. **European Journal of Spatial Development**, n. 12, p. 1-20, 2004.

WANG, H.; Chen, B.; He, X.; Tong, Q.; Zhao, J. SDG-based HAZOP analysis of operating mistakes for PVC process. **Process Safety and Environmental Protection**, v. 87, n. 1, 40–46, 2009.

WANG, Y.-F.; WU, J.-Y.; CHANG, C.-T. Automatic hazard analysis of batch operations with Petri nets. **Reliability Engineering & System Safety**, v. 76, n. 1, p. 91–104, 2002.

WOODCOCK, B.; AU, Z. Human factors issues in the management of emergency response at high hazard installations. **Journal of Loss Prevention in the Process Industries**, v. 26, p. 547-557, 2013.

ZALDÍVAR, J. M.; CANO, J.; ALÓS, M. A.; SEMPERE, J.; NOMEN, R.; LISTER, D.; MASCHIO, G.; OBERTOPP, T.; GILLES, E. D.; BOSCH, J.; STROZZI, F. A general criterion to define runaway limits in chemical reactors. **Journal of Loss Prevention in the Process Industries**, v. 16, n. 3, p. 187–200, 2003.

Apêndice A

Códigos do MATLAB

```

      ROTINA PRINCIPAL
1  %% main
2  %===== main =====
3  % Rotina responsável pela chamada de todos os módulos do algoritmo da
4  % dissertação
5  %=====
6  close all; clear all; pause off; format short; clc
7  %=====
8  % Definindo parâmetros da simulação e do NMPC
9  C = 3;           % Tamanho do horizonte de controle (M na dissertação)
10 P = 10;          % Tamanho do horizonte de predição
11 d = 1;           % Contador. Não mude
12 N = 100;         % Número total de ações simuladas
13 FA = 20;         % Converte ii -> tempo [h] (Ta em minutos é 60/FA)
14 hrec = 8;        % Horizonte de identificação
15 Hcvg = 60;       % Horizonte de simulação considerado no Calcseg
16 stdv = 0.4;      % Desvio padrão do ruído
17 modseg = 0;      % Ativa módulo de segurança
18 recon = 1;       % Ativa reconciliação
19 testcontr = 1;   % Modo de teste
20 simfalha = 0;    % Simula falha em 4 h
21 falha = 'n';     % Tipo de falha em estudo: UA ou Fincat ou Tc
22 SaveName = 'Nome.mat'; % Nome para salvar o workspace
23 %=====
24 % Definindo os parâmetros do sistema
25 Fin = 112.69;    % [mol/h]
26 Fins = 198.62;  % [mol/h]
27 Fincat = 0.0993; % [mol/h]
28 Tc = 361;       % [K]
29 hA = 1;         % [1/h]
30
31 Fincatfail = 1.5; % Falha: if próximo à integração da planta no main.m
32 hAfail = 0.1;    % Falha: if dentro do Planta.m
33 Tcfail = 375;   % Falha: if no cabeçalho do Controlador.m
34
35 Tin = 350;      % [K]
36 k0 = 5e7;       % [1/h]
37 MM = 60/1000;   % [kg/mol]
38 Ea = 2400;      % [cal/mol]
39 RR = 1.987;     % [cal/mol.K]
40 dH = 6000/1000; % [kcal/mol]

```

```

41 Cp = 0.5*MM;           % MM*Cp [kcal/mol]
42
43 Tlim = 385;           % Limite de temperatura
44
45 % Definindo os pesos do controlador MIMO
46 param.wQ = [1 0 1 1 100]; %Mon Pol T Sol Cat
47 param.wR = [1; 100; 0.5; 0.25]; %Tc Fincat Fin Fins
48 param.wS = [1; 100; 0.5; 0.25]; %Tc Fincat Fin Fins
49 param.WQ = 1;         % Ativa peso Q
50 param.WR = 1;         % Ativa peso R
51 param.WS = 0;         % Ativa peso S
52
53 % Construindo o struct parmod que alimenta o modelo e a planta
54 parmod.Fin = Fin;
55 parmod.Fins = Fins;
56 parmod.Fincat = Fincat;
57 parmod.Tin = Tin;
58 parmod.k0 = k0;
59 parmod.Ea = Ea;
60 parmod.RR = RR;
61 parmod.dH = dH;
62 parmod.Cp = Cp;
63 parmod.hA = hA;
64 parmod.hAplant = hA;
65 parmod.Fincatfail = Fincatfail;
66 parmod.hAfail = hAfail;
67 parmod.Tcfail = Tcfail;
68 parmod.falha = falha;
69 parmod.simfalha = simfalha;
70
71 %=====
72 % Chutes iniciais do controlador
73 Tc0 = Tc*ones(1,C);
74 Fc0 = Fincat*ones(1,C);
75 Fin0 = Fin*ones(1,C);
76 Fs0 = Fins*ones(1,C);
77 u0 = [Tc0 Fc0 Fin0 Fs0];
78 u01 = [Tc0 Fc0 Fin0 Fs0];
79 %=====
80 % Definindo os limites das ações de controle
81 Tcu = 420;           % Temperatura máxima da camisa (K)
82 Tcl = 300;           % Temperatura mínima da camisa (K)
83 Fcu = 2;             % Vazão máxima de catalisador na alimentação (kg/h)
84 Fcl = 0;             % Vazão mínima de catalisador na alimentação (kg/h)
85 Finu = 200;
86 Finl = 0;
87 Fsu = 400;
88 Fsl = 0;
89 %=====
90 % Limites das variáveis de controle: lb -> Lower Bound;  ub -> Upper Bound
91 lbTc = Tcl*ones(1,C);
92 ubTc = Tcu*ones(1,C);
93 lbFc = Fcl*ones(1,C);
94 ubFc = Fcu*ones(1,C);
95 lbFin = Finl*ones(1,C);
96 ubFin = Finu*ones(1,C);
97 lbFs = Fsl*ones(1,C);
98 ubFs = Fsu*ones(1,C);

```

```

99 param.ContrBounds = struct('lbTc',Tcl,'ubTc',Tcu,'lbFc',Fcl,'ubFc',Fcu,...
100                          'lbFin',Finl,'ubFin',Finu,'lbFs',Fsl,'ubFs',Fsu);
101 % Colocando tudo no mesmo vetor..
102 lb = [lbTc Fc0 Fin0 Fs0];
103 ub = [ubTc Fc0 Fin0 Fs0];
104 lbseg = [lbTc lbFc lbFin lbFs];
105 ubseg = [ubTc ubFc ubFin ubFs];
106 lbrec = 0;
107 ubrec = 2;
108 %=====
109 % Configurando o otimizador
110 % Controlador
111 opContr = optimset('Algorithm','interior-point','UseParallel','always',...
112                  'display','off','TolX',1e-6,'TolFun',1e-7,...
113                  'TolCon',1e-5,'MaxFunEvals',4000);
114 % Reconciliador
115 opRecon = optimset('Algorithm','interior-point','UseParallel','always',...
116                  'display','off','TolX',1e-8,'TolFun',1e-9);
117 %=====
118 % Condição inicial do sistema
119 CIsis = [36.2225 77.2474 380.0101 200.0000 0.1000];
120 x0contr = CIsis; % Condição inicial do controlador
121 x0plant = CIsis; % Condição inicial da planta
122
123 neq = length(CIsis);
124 histxrecon = zeros(N,neq); % Armazena os estados reconciliados
125 histVM = zeros(N+1,1); % Armaz as Variáveis Medidas da planta
126 histx0plant = zeros(N+1,neq); % Armaz os est verd da planta (sem ruído)
127 histtplant = zeros(N+1,1); % Armaz os tempos de amost da planta
128 histTc = zeros(N+1,1); % Armaz as atuações em Tc
129 histFc = zeros(N+1,1); % Armaz as atuações em Fincat
130 histFin = zeros(N+1,1); % Armaz as atuações em Fin
131 histFins = zeros(N+1,1); % Armaz as atuações em Fins
132 histhA = zeros(N+1,1); % Armaz os valores de hA do sistema
133 TempoIteracao = zeros(N,1); % Armaz o tempo gasto em cada iteração
134 % histVM2 = zeros(N+1,neq);
135
136 histVM(1) = CIsis(3);
137 histx0plant(1,:) = x0plant;
138 histtplant(1) = 0;
139 histTc(1) = Tc;
140 histFc(1) = Fincat;
141 histFin(1) = Fin;
142 histFins(1) = Fins;
143 histhA(1) = hA;
144 % histVM2(1,:) = CIsis;
145 % VM2 = x0plant;
146
147 %=====
148 % Construindo o struct param
149 param.N = N;
150 param.d = d;
151 param.C = C;
152 param.P = P;
153 param.FA = FA;
154 param.u01 = u01;
155 param.u0 = u0;
156 param.lb = lb;

```



```

157 param.ub = ub;
158 param.lbrec = lbrec;
159 param.ubrec = ubrec;
160 param.lbseg = lbseg;
161 param.ubseg = ubseg;
162 param.opContr = opContr;
163 param.opRecon = opRecon;
164 param.histTc = histTc;
165 param.histFc = histFc;
166 param.histFin = histFin;
167 param.histFins = histFins;
168 param.hrec = hrec;
169 param.neq = neq;
170 param.Hcvg = Hcvg;
171 param.falha = falha;
172 param.recon = recon;
173 param.testcontr = testcontr;
174 param.simfalha = simfalha;
175 param.Tlim = Tlim;
176 param.simse = 0;
177 param.modseg = modseg;
178 param.FeedEE0 = [Tc Fincat Fin Fins];
179
180 flagseg = 0;      % Contador
181 % Define a configuração inicial do controlador: SISO(0) MIMO(1)
182 prisc = 0;
183 % Define o setpoint inicial do controlador
184 SPseg = [36.2225   77.2465  380.0101  200.0000   0.1000];
185
186 % Iniciando a simulação
187 for ii = 1:1:N;      % Número de movimentos da simulação
188     tempproc = cputime;
189
190     % if ii == 20
191     % % SPseg = [25   114.7730  360.0101  210   0.3000];%MIMO não viável
192     % SPseg = [22.2259  104.6468  362.7408  190.6180   0.2723];%MIMO viável
193     % param.FeedEE = [308.5805; 0.2792; 130.0825; 195.4406];
194     % % SPseg = [36.2225   77.2465  360.0101  200   0.1000];%SISO
195     % prisc = 1;
196     % end
197
198
199 %=====
200 %                               MÓDULO DE RECONCILIAÇÃO
201 %=====
202 if recon
203     if ii>hrec, temprec = cputime;
204         u0rec = histhA(ii);
205         [hAcalc] = fmincon(@(u0rec)Reconc(u0rec,ii,histxrecon,histVM,...
206                                 param,paramod),u0rec,[],[],[],[],lbrec,ubrec,...
207                                 [],opRecon);
208         disp(['hA calculado = ' mat2str(hAcalc,4)])
209         [~,xrecon] = Reconc(hAcalc,ii,histxrecon,histVM,param,paramod);
210         paramod.hA = hAcalc;
211         TempoRecon(ii-hrec) = cputime-temprec;
212     else
213         hAcalc = hA;
214         xrecon = x0plant;

```

```

215         parmod.hA = hAcalc;
216     end
217     x0contr = xrecon;
218     histxrecon(ii,:) = xrecon;
219     histhA(ii+1,1) = hAcalc;
220 else
221     x0contr = x0plant;
222     %     x0contr = VM2;
223 end
224
225 %=====
226 %                               MÓDULO DE SEGURANÇA
227 %=====
228
229 % O módulo de segurança (modseg) atua de tempos em tempos amostrando os
230 % estados da planta (x0plant) e simulando acidentes. Caso um desses
231 % acidentes supere um determinado limite segurança de T/P o 'calcseg' é
232 % acionado de modo a encontrar o estado mais próximo no qual o acidente
233 % não ultrapasse o limite de segurança.
234
235 if modseg && ~prisc && ii == 20           % Ativa o SimSeg em 1 h de operação
236     disp('SIMULAÇÃO DE SEGURANÇA INICIADA')
237     tempsimseg = cputime;
238     [simsegOUT,prisc] = simseg(ii,x0contr,SPseg,u0,param,parmod,...
239                             histVM(1:ii),histhA(1:ii+1),...
240                             histxrecon(1:ii,:));
241     TempoSimSeg = cputime-tempsimseg;
242     if prisc
243         disp('MODO DE SEGURANÇA ATIVADO: CALCULANDO NOVOS SETPOINTS')
244         tempcalcseg = cputime;
245         [SPseg,calcsegOUT] = calcseg(x0contr,ii,parmod,param,...
246                                   histhA(ii-4:ii+1));
247
248         disp(['FeedEE = ' mat2str(calcsegOUT.FeedEE,4)])
249         param.FeedEE = calcsegOUT.FeedEE;
250         TempoCalcSeg = cputime-tempcalcseg;
251     end
252 end
253
254 %=====
255 %                               MÓDULO DE CONTROLE
256 %=====
257 [TFcalc] = Controlador(u0,ii,x0contr,prisc,SPseg,param,parmod);
258 % Separando as ações de controle
259 Tcvec = TFcalc(1:C);
260 Fcvec = TFcalc(C+1:2*C);
261 Finvec = TFcalc(2*C+1:3*C);
262 Finsvec = TFcalc(3*C+1:4*C);
263
264 %=====
265 %                               MÓDULO DA PLANTA
266 %=====
267 % Atualizando as variáveis manipuladas
268 parmod.Tc = Tcvec(1);
269 parmod.Fincat = Fcvec(1);
270 parmod.Fin = Finvec(1);
271 parmod.Fins = Finsvec(1);
272

```

```

273 % if ii>=80, parmod.Tc = histTc(80); end %
274
275 % Aplica falha na planta
276 if simfalha && ii == 80
277     modseg = 0;
278     param.modseg = modseg;
279     prisc = 0;
280     parmod.Fin = histFin(80);
281     parmod.Fins = histFins(80);
282
283     if strcmp(param.falha,'Fincatn'), parmod.Fincat = parmod.Fincatfail;
284     else parmod.Fincat = histFc(80); end
285
286     param.lb(C+1:4*C) = [parmod.Fincat*ones(1,C), parmod.Fin*ones(1,C),...
287                         parmod.Fins*ones(1,C)];
288     param.ub(C+1:4*C) = [parmod.Fincat*ones(1,C), parmod.Fin*ones(1,C),...
289                         parmod.Fins*ones(1,C)];
290 end
291 % Aplica a primeira ação de controle calculada na planta
292 tempo=((ii-1):d:ii)/FA;
293 [tplant xplant]=dasslc('Planta',tempo,x0plant,[],parmod);
294
295 %=====
296 %           MÓDULO DE ATUALIZAÇÃO E ARMAZENAMENTO DE VARIÁVEIS
297 %=====
298 % Atualização do vetor de ações de controle de acordo com deslocamento
299 % do horizonte
300 u0 = [Tcvec(2:C) Tcvec(C) Fcvec(2:C) Fcvec(C)...
301       Finvec(2:C) Finvec(C) Finsvec(2:C) Finsvec(C)];
302 % A resposta final da planta torna-se a nova condição inicial da planta
303 x0plant = xplant(end,:);
304
305 % Corrompendo a variável medida com ruído normal
306 VM = x0plant(3) + normrnd(0, stdv);
307 % VM2 = x0plant + normrnd(zeros(1,5), x0plant*0.01);
308 % VM2(3) = VM;
309 % if ii>=80, VM = histx0plant(80,3) + normrnd(0, stdv); end%
310
311 % Armazenando a resposta da planta e as ações de controle calculadas
312 histx0plant(ii+1,:) = x0plant;
313 histVM(ii+1,1) = VM;
314 % histVM2(ii+1,:) = VM2;
315 histtplant(ii+1,1) = tplant(end);
316 histTc(ii+1,1) = parmod.Tc;
317 histFc(ii+1,1) = parmod.Fincat;
318 histFin(ii+1,1) = parmod.Fin;
319 histFins(ii+1,1) = parmod.Fins;
320 param.histTc = histTc;
321 param.histFc = histFc;
322 param.histFin = histFin;
323 param.histFins = histFins;
324
325 TempoIteracao(ii) = cputime-tempproc;
326
327 %=====
328 %           MÓDULO DE INTERAÇÃO COM O USUÁRIO
329 %=====
330 % Informações mostradas na tela ao final de cada processo de otimização

```

```

331 disp(['Finalizada a iteração ii = ' mat2str(ii)])
332 disp(['Temperatura da camisa = ' mat2str(parmod.Tc,4)])
333 disp(['Fin = ' mat2str(parmod.Fin,4)])
334 disp(['Fins = ' mat2str(parmod.Fins,4)])
335 disp(['Fincat = ' mat2str(parmod.Fincat,4)])
336 disp(['Setpoint = ' mat2str(SPseg,4)])
337 disp(['Estados estimados: ' mat2str(histxrecon(ii,:),4)])
338 disp(['Tempo de Iteração = ' mat2str(TempoIteracao(ii),3) ' s'])
339 disp(' ')
340 %=====
341
342 end
343
344 TempoTotal = sum(TempoIteracao)/60;
345 disp(['TempoTotal: ' mat2str(TempoTotal,3) ' min'])
346 % TempoReconMed = mean(TempoRecon);
347
348 %=====
349 %
350 %
351
352 save(SaveName)
353
354 % param.FeedEE0 = [112.69 198.62 0.0993 361];
355 % param.FeedEE = [308.5805; 0.2792; 130.0825; 195.4406]; % só pra funfar..
356 % PlotTestContr(histtplant,histx0plant,FA,CIsis,SPseg,ii,histTc,histFin,...
357 % histFins,histFc,param.FeedEE0,param.FeedEE,parmod.hAplant,histhA)
358 %
359
360 % PlotTestReconc(histtplant,histxrecon,FA,CIsis,SPseg,ii,histTc,histFin,...
361 % histFins,histFc,param.FeedEE0,param.FeedEE,parmod.hAplant,histhA)
362
363
364 % PlotSimCompIns(histtplant,histxrecon,FA,CIsis,SPseg,ii,histhA,histTc)
365
366
367 %%
368 %===== Construção do Gráfico de Estados =====
369 close all
370 Xlim = [0 5];
371 XTick = [0:1:Xlim(2)];
372 Loc = {'Northeast','Southeast','East'};
373 Locjj = {Loc{2},Loc{1},Loc{2},Loc{2},Loc{1}};
374 lab = {'Monômero','Polímero','Temperatura','Solvente','Catalisador'};
375 Ylim = {[30 45],[70 85],[370 400],[180 210],[0 0.4]};
376 Locjj2 = {Loc{2},Loc{2},Loc{2},Loc{1}};
377 lab2 = {'Monômero','Temperatura','Solvente','Catalisador'};
378 Ylim2 = {[0 200],[300 370],[0 400],[0 1]};
379
380 fg.LW = 2;
381 fg.LW2 = 4;
382 fg.MS = 10;
383 fg.MS2 = 8;
384 fg.FS = 30;
385 fg.FS2 = 26;
386 fg.c1 = [0 0 0];
387 fg.c2 = [.5 .5 .5];
388 fg.c3 = [1 1 1];

```

```

389
390 % ssf = [simsegOUT.histFin,simsegOUT.histTc,simsegOUT.histFins,...
391 %       simsegOUT.histFc];
392 % sshxrc = simsegOUT.histxrecon;
393 % sshtpl = simsegOUT.histtplant;
394 % sshhA = simsegOUT.histhA(1:end-1);
395 % sslt = length(sshtpl);
396 hf = [histFin,histTc,histFins,histFc];
397 htpl = histtplant;
398 % hxrc = histVM2;
399 hxrc = histxrecon;
400 % tproj = [sshtpl(1):1/FA:htpl(end)]';
401 tproj = [1:1/FA:htpl(end)]';
402 if exist('calcsegOUT','var')
403     csx = calcsegOUT.csx;
404     cst = calcsegOUT.cst + 3; % o '+3' é p/ plot junto c/ falha da planta
405     csf = [calcsegOUT.csparmod.Fin',calcsegOUT.csparmod.Tc',...
406           calcsegOUT.csparmod.Fins',calcsegOUT.csparmod.Fincat'];
407     cshhA = calcsegOUT.csparmod.hA;
408     ts = 0:1/FA:cst(end);
409 end
410 %%
411 % Gráficos variáveis controladas
412 for jj = 3 %1:5
413 figure(jj);
414 hold on;
415 % resp5=stairs(htpl,[CIsis(jj)*ones(FA,1); SPseg(jj)*ones(length(htpl)-FA,1)],...
416 %   '--','LineWidth',fg.LW2,'Color',fg.c1);
417 if jj==3,
418 resp5=stairs(htpl,[CIsis(jj)*ones(FA,1); SPseg(jj)*ones(length(htpl)-FA,1)],...
419 %   '--','LineWidth',fg.LW2,'Color',fg.c1);
420 % resp5=plot(htpl,Tlim*ones(size(htpl)),'k-','LineWidth',fg.LW,'Color',fg.c1);
421 resp6=plot(htpl(2:ii+1),histVM(1:ii),...
422 %   'ko','MarkerFaceColor',fg.c3,'MarkerSize',fg.MS);
423 end
424 % resp3=plot(tproj,SPseg(jj)*ones(size(tproj)),...
425 %   'k--','LineWidth',fg.LW,'Color',fg.c1);
426 % resp2=plot(sshtpl,sshxrc(end-sslt+1:end,jj),...
427 %   'kx','MarkerFaceColor',fg.c2,'MarkerSize',fg.MS,'LineWidth',fg.LW);
428 % resp4=plot(cst,csx(:,jj),'kd','MarkerFaceColor',fg.c2,'MarkerSize',fg.MS);
429 resp =plot(htpl(2:ii+1),hxrc(1:ii,jj),...
430 %   'ko','MarkerFaceColor',fg.c1,'MarkerSize',fg.MS);
431 % resp6=plot(htpl(2:ii+1),histx0plant(1:ii,jj),...
432 %   'ks','MarkerFaceColor',fg.c3,'MarkerSize',fg.MS2);
433 hold off;
434 set(gca,'FontSize',fg.FS,'XLim',Xlim,'YLim',Ylim{jj},'box','on','XTick',...
435 %   XTick,'LineWidth',fg.LW)
436 xlabel('Tempo [h]','FontSize',fg.FS,'Position',...
437 %   [mean(get(gca,'XLim')) min(get(gca,'YLim'))-0.07*diff(get(gca,'YLim')) 1]);
438 if jj==3
439 %   legend([resp;resp6;resp2;resp4;resp3;resp5],'Resp Estim','Resp Real',...
440 %   %   'Sim Falha','Resp Predita','SPseg','Limite Seg');
441 %   set(legend,'FontSize',fg.FS2,'Location',Locjj{jj});
442 %   ylabel([lab{jj} ' do reator [K]'],'FontSize',fg.FS,'Position',...
443 %   %   [-0.7 mean(get(gca,'YLim')) 1]); %0.55
444 %   legend([resp;resp5],'Resposta','Setpoint');
445 %   legend([resp;resp6;resp5],'Estimado','Medido','Setpoint');
446 %   legend([resp;resp2;resp4;resp3;resp5],'Resp Estim',...

```

```

447 %           'Sim Falha','Resp Predita','SPseg','Limite Seg');
448 set(legend,'FontSize',fg.FS2,'Location',Locjj{jj});
449 ylabel([lab{jj} ' do reator [K]'],'FontSize',fg.FS,'Position',...
450         [-0.35 mean(get(gca,'YLim')) 1]); %0.55 0.7
451 else
452 % % %       legend([resp;resp6;resp2;resp4;resp3],'Resp Estim','Resp Real',...
453 % % %           'Sim Falha','Resp Predita','SPseg');
454 % % %       set(legend,'FontSize',fg.FS,'Location',Locjj{jj});
455 % % %       ylabel([lab{jj} ' no reator [mol]'],'FontSize',fg.FS,'Position',...
456 % % %           [-0.7 mean(get(gca,'YLim')) 1]); %0.55
457 %
458 %           legend([resp;resp5],'Resposta','Setpoint');
459 % %         legend([resp;resp2;resp4;resp3],'Resp Estim',...
460 % %             'Sim Falha','Resp Predita','SPseg');
461 % %         set(legend,'FontSize',fg.FS,'Location',Locjj{jj});
462 ylabel([lab{jj} ' no reator [mol]'],'FontSize',fg.FS,'Position',...
463         [-0.35 mean(get(gca,'YLim')) 1]); %0.55
464 end
465 drawnow;
466 jFrame = get(handle(gcf),'JavaFrame');
467 jFrame.setMaximized(true);
468 end
469 % return
470 %% Gráficos variáveis manipuladas
471 for jj = 2 %1:4
472 figure(jj+5);
473 hold on;
474 % resp2=stairs(sshtpl,ssf(end-sslt+1:end,jj),...
475 %             'k--x','MarkerFaceColor',fg.c1,...
476 %             'MarkerSize',fg.MS,'LineWidth',fg.LW2,'Color',fg.c2);
477 % resp4=stairs(cst,csf(:,jj),'k-', 'MarkerFaceColor',fg.c2,'MarkerSize',...
478 %             fg.MS,'LineWidth',fg.LW2,'Color',fg.c2);
479 resp =stairs(htpl(2:ii+1),hf(1:ii,jj),'k-', 'MarkerFaceColor',fg.c1,...
480             'MarkerSize',fg.MS,'LineWidth',fg.LW2);
481 % if jj==4, resp5 =stairs(htpl(2:ii+1),hr(1:ii,4),
482 %             'k--','MarkerFaceColor',...
483 %             fg.c3,'MarkerSize',fg.MS2,'LineWidth',fg.LW2);end
484 hold off;
485 set(gca,'FontSize',fg.FS,'XLim',Xlim,'YLim',Ylim2{jj},'box','on',...
486       'XTick',XTick,'LineWidth',fg.LW)
487 if jj==2
488     ylabel([lab2{jj} ' da camisa [K]'],'FontSize',fg.FS,'Position',...
489           [-0.35 mean(get(gca,'YLim')) 1]); %0.55
490 else
491     ylabel(['Vazão de ',lab2{jj},' [mol/h]'],'FontSize',fg.FS,...
492           'Position',[-0.35 mean(get(gca,'YLim')) 1]); %0.55 0.7
493 end
494 xlabel('Tempo [h]','FontSize',fg.FS,'Position',...
495 [mean(get(gca,'XLim')) min(get(gca,'YLim'))-0.07*diff(get(gca,'YLim')) 1]);
496 % % if jj==4
497 % % legend([resp;resp5;resp2;resp4],'Resp Contr','Resp Real','Sim Falha',...
498 % %     'Resp Predita'); set(legend,'FontSize',fg.FS,'Location',Locjj2{jj});
499 % % else
500 % legend([resp;resp2;resp4],'Resposta','Sim Falha','Resp Predita');
501 % set(legend,'FontSize',fg.FS,'Location',Locjj2{jj});
502 % % end
503 drawnow;
504 jFrame = get(handle(gcf),'JavaFrame');

```

```

505 jFrame.setMaximized(true);
506 end
507
508 % return
509 %%
510 % Gráfico do UA
511 hAplant=1;
512 hAplant2 = 0.1+0.9*exp(-(1:0.05:6)-1)');
513 % hAplant2 = 0.1*ones(ii-1*FA,1);
514 SPhAplant = [hAplant*ones(1*FA+1,1); hAplant2(2:end)];
515 figure(10);
516 hold on;
517 % resp2=plot(sshtpl,sshhA(end-sslt+1:end,1),'kx','MarkerFaceColor',fg.c2,...
518 % 'MarkerSize',fg.MS,'LineWidth',fg.LW);
519 % resp4=plot(cst,cshhA,'kd','MarkerFaceColor',fg.c2,'MarkerSize',fg.MS);
520 resp=plot(htpl(1:ii),histhA(1:ii,1),'ko','MarkerFaceColor',fg.c1,...
521 'MarkerSize',fg.MS);
522 resp3=plot(htpl(1:ii),SPhAplant(1:ii),'ko','MarkerFaceColor',fg.c3,...
523 'MarkerSize',fg.MS);
524 hold off;
525 set(gca,'FontSize',fg.FS,'XLim',Xlim,'YLim',[0 1.2],'box','on','XTick',...
526 XTick,'LineWidth',fg.LW);
527 ylabel('\kappa_c [1/h]','FontSize',fg.FS,...
528 'Position',[-0.35 mean(get(gca,'YLim')) 1]); %0.7
529 xlabel('Tempo [h]','FontSize',fg.FS,'Position',...
530 [mean(get(gca,'XLim')) min(get(gca,'YLim'))-0.07*diff(get(gca,'YLim')) 1]);
531 % legend([resp;resp2;resp4],'Resposta','Sim Falha','Resp Predita');
532 legend([resp;resp3],'Estimado','Real');
533 set(legend,'FontSize',fg.FS,'Location',Loc{1});
534 drawnow;
535 JFrame = get(handle(gcf),'JavaFrame');
536 JFrame.setMaximized(true);
537
538 return
539 %% Gráficos de Conversão e Produção de Polímero
540 % Mt = histxrecon(:,1)+histxrecon(:,2)+histxrecon(:,4)+histxrecon(:,5);
541 Mt = histx0plant(:,1)+histx0plant(:,2)+histx0plant(:,4)+histx0plant(:,5);
542 % xMon = histxrecon(:,1)./Mt;
543 xMon = histx0plant(:,1)./Mt;
544 Fliq = histFin + histFc + histFins;
545 % FliqMon = xMon.*Fliq(1:end-1);
546 FliqMon = xMon.*Fliq;
547 % X = (histFin(1:end-1)-FliqMon)./histFin(1:end-1);
548 X = (histFin-FliqMon)./histFin;
549
550 figure(11);
551 plot(htpl(1:ii),X(1:ii),'ko','MarkerFaceColor',fg.c1,'MarkerSize',fg.MS);
552 set(gca,'FontSize',fg.FS,'YLim',[0.4 0.8],'XLim',Xlim,'box','on','XTick',...
553 XTick,'LineWidth',fg.LW);
554 ylabel('Conversão','FontSize',fg.FS,...
555 'Position',[-0.7 mean(get(gca,'YLim')) 1]);
556 xlabel('Tempo [h]','FontSize',fg.FS,'Position',...
557 [mean(get(gca,'XLim')) min(get(gca,'YLim'))-0.07*diff(get(gca,'YLim')) 1]);
558 drawnow;
559 JFrame = get(handle(gcf),'JavaFrame');
560 JFrame.setMaximized(true);
561
562 % xPol = histxrecon(:,2)./Mt;

```

```

563 xPol = histx0plant(:,2)./Mt;
564 % FliqPol = xPol.*Fliq(1:end-1);
565 FliqPol = xPol.*Fliq;
566
567 figure(12);
568 plot(htpl(1:ii),FliqPol(1:ii),'ko','MarkerFaceColor',fg.cl,...
569      'MarkerSize',fg.MS);
570 set(gca,'FontSize',fg.FS,'YLim',[60 140],'XLim',Xlim,'box','on','XTick',...
571      XTick,'LineWidth',fg.LW);
572 ylabel('Produção de Polímero [mol/h]','FontSize',fg.FS,'Position',...
573      [-0.7 mean(get(gca,'YLim')) 1]);
574 xlabel('Tempo [h]','FontSize',fg.FS,'Position',...
575      [mean(get(gca,'XLim')) min(get(gca,'YLim'))-0.07*diff(get(gca,'YLim')) 1]);
576 drawnow;
577 jFrame = get(handle(gcf),'JavaFrame');
578 jFrame.setMaximized(true);

```

RECONCILIADOR

```

1 function [FO, xrecon] = Reconc(u0rec,ii,histxrecon,histVM,param,parmod)
2
3 d = param.d;
4 hrec = param.hrec;
5 FA = param.FA;
6 Tc = param.histTc(ii+1-hrec:ii); % Tamanho Hrec+1
7 Fc = param.histFc(ii+1-hrec:ii); % Tamanho Hrec+1
8 Fin = param.histFin(ii+1-hrec:ii); % Tamanho Hrec+1
9 Fins = param.histFins(ii+1-hrec:ii); % Tamanho Hrec+1
10 parmod.hA = u0rec;
11 tempo = ((ii-hrec-1):d:(ii-1))/FA; % Tamanho Hrec
12
13 xr3 = zeros(1,hrec+1);
14 xr3(1) = histxrecon(ii-hrec,3);
15
16 xini = histxrecon(ii-hrec,:); % ii=6 ; xini=histxrecon(1)
17 for jj = 1:hrec
18     parmod.Tc = Tc(jj);
19     parmod.Fincat = Fc(jj);
20     parmod.Fin = Fin(jj);
21     parmod.Fins = Fins(jj);
22     tempo2 = [tempo(jj) tempo(jj+1)];
23
24     [~,xr]=dasslc('ModeloContr',tempo2,xini,[],parmod);
25     xini = xr(end,:);
26
27     xr3(jj+1) = xr(end,3);
28 end
29 xrecon = xr(end,:);
30
31 xm = histVM(ii-hrec:ii);
32 erro = xr3' - xm;
33
34 FO = erro'*erro;
35

```

CONTROLADOR

```

1 %=====
2 %                               Controlador
3 %=====

```



```

4 function [TFcalc] = Controlador(u0,ii,x0contr,prisc,SPseg,param,parmod)
5
6 if prisc
7     u0n = (u0 - param.lbseg)./(param.ubseg - param.lbseg);
8     [TFcalcn,~,ResultContr] = fmincon(@(u0n)ContrSeg(u0n,ii,x0contr,...
9         SPseg,param,parmod),u0n,[],[],[],[],zeros(size(u0n)),...
10        ones(size(u0n)),...
11        @(u0n)RestrContr(u0n,ii,param.C,param.lbseg,param.ubseg,...
12        param.histTc(ii),param.u0l(1)),param.opContr);
13     TFcalc = TFcalcn.*(param.ubseg - param.lbseg) + param.lbseg;
14 else
15     if param.simfalha && ~param.simse && ii>= 80 && strcmp(param.falha,'Tc')
16         TFcalc = u0;
17         TFcalc(1:param.C) = parmod.Tcfail;
18     elseif param.simse && strcmp(param.falha,'Tc')
19         TFcalc = u0;
20         TFcalc(1:param.C) = parmod.Tcfail;
21     else
22         u0n = u0;
23         [TFcalcn,~,ResultContr,out] = fmincon(@(u0n)Contr(u0n,ii,x0contr,...
24             SPseg,param,parmod),u0n,[],[],[],[],param.lb,param.ub,...
25             @(u0n)RestrContr(u0n,ii,param.C,param.lbseg,...
26             param.ubseg,param.histTc(ii),param.u0l(1)),param.opContr);
27         TFcalc = TFcalcn;
28     end
29 end
30
31 if exist('ResultContr','var')
32 disp('ResultContr')
33 if ResultContr == -2
34     disp('No feasible point found')
35 elseif ResultContr == 0
36     disp('Too many function evaluations or iterations')
37 elseif ResultContr == 1
38     disp('First order optimality conditions satisfied')
39 else
40     disp(ResultContr)
41 end
42 end
43
44
45 %=====
46 %             Função objetivo do Controlador
47 %=====
48 function FO = Contr(u0n,ii,x0contr,SPseg,param,parmod)
49
50 u0 = u0n;
51 P = param.P;
52 d = param.d;
53 C = param.C;
54 FA = param.FA;
55
56 xr3 = zeros(1,P+1);
57 xr3(1) = x0contr(3);
58
59 xini = x0contr;
60 tempo=((ii-1):d:(P+ii-1))/FA;    %horizonte de predição se move
61 for jj = 1:C    % mantém a ação de contr de modo contínuo de ii até ii+1

```

```

62     parmod.Tc = u0(jj);
63
64     if jj==C,     tempo2 = [tempo(jj):d/FA:tempo(end)];
65     else         tempo2 = [tempo(jj) tempo(jj+1)];     end
66
67     [~,xr]=dassl('ModeloContr',tempo2,xini,[],paramod);
68     xini = xr(end,:);
69
70     if jj==C,     xr3(jj+1:end) = xr(2:end,3)';
71     else         xr3(jj+1) = xr(end,3);     end
72 end
73
74 % Calculando os erros para compor a função objetivo
75 erro1 = xr3- SPseg(3);
76 erro1q = erro1*erro1';
77
78 % Calculando a diferença entre as ações de controle do vetor u
79 delta = zeros(1,C-1);
80 delta(1:C-1) = u0((1:C-1)+1)-u0(1:C-1);
81 erro2q = delta(1:C-1)*delta(1:C-1)'; % delta entre as ações de contr k e k+1
82
83 FO = erro1q*param.WQ + erro2q*param.WR;
84
85 %=====
86 %                               Restrições do Controlador
87 %=====
88 function [ c, ceq ] = RestrContr(u0n,ii,C,lbseg,ubseg,histTc,u01)
89
90 % No controle SISO não há normalização pq é só a Temperatura (TRUE)
91 % Mas no controle MIMO há normalização! (FALSE)
92 if u0n(1) > 299,     u0 = u0n;
93 else                u0 = u0n(1:C).*(ubseg(1:C) - lbseg(1:C)) + lbseg(1:C);
94 end
95
96 % Construindo as restrições na forma C(X) <= 0
97 % \Delta{Tc} <= 5 K
98 % Para o vetor u da otimização
99 c(1:C-1) = abs(u0((1:C-1)+1)-u0(1:C-1)) - 5;
100
101 % Para considerar a última ação aplicada na planta
102 if ii==1,     c(C) = abs(u0(1)-u01) - 5;
103 else         c(C) = abs(u0(1)-histTc) - 5;     end
104
105 ceq = [];
106
107
108
109
110 %=====
111 %                               Função objetivo do Controlador para prisc = 1
112 %=====
113 function FO = ContrSeg(u0n,ii,x0contr,SPseg,param,paramod)
114
115 u0 = u0n.*(param.ubseg - param.lbseg) + param.lbseg;
116
117 P = param.P;
118 d = param.d;
119 C = param.C;

```

```

120 FA = param.FA;
121
122 xr1 = zeros(1,P+1);
123 xr2 = zeros(1,P+1);
124 xr3 = zeros(1,P+1);
125 xr4 = zeros(1,P+1);
126 xr5 = zeros(1,P+1);
127
128 xr1(1) = x0contr(1);
129 xr2(1) = x0contr(2);
130 xr3(1) = x0contr(3);
131 xr4(1) = x0contr(4);
132 xr5(1) = x0contr(5);
133
134 xini = x0contr;
135 tempo=((ii-1):d:(P+ii-1))/FA;      %horizonte de predição se move
136 for jj = 1:C
137     parmod.Tc = u0(jj);
138     parmod.Fincat = u0(jj+C);
139     parmod.Fin = u0(jj+2*C);
140     parmod.Fins = u0(jj+3*C);
141
142     if jj==C
143         tempo2 = [tempo(jj):d/FA:tempo(end)];
144     else
145         tempo2 = [tempo(jj) tempo(jj+1)];
146     end
147
148     [~,xr]=dassl('ModeloContr',tempo2,xini,[],parmod);
149     xini = xr(end,:);
150
151     if jj==C
152         xr1(jj+1:end) = xr(2:end,1)';
153         xr2(jj+1:end) = xr(2:end,2)';
154         xr3(jj+1:end) = xr(2:end,3)';
155         xr4(jj+1:end) = xr(2:end,4)';
156         xr5(jj+1:end) = xr(2:end,5)';
157     else
158         xr1(jj+1) = xr(end,1);
159         xr2(jj+1) = xr(end,2);
160         xr3(jj+1) = xr(end,3);
161         xr4(jj+1) = xr(end,4);
162         xr5(jj+1) = xr(end,5);
163     end
164 end
165 xr = [xr1;xr2;xr3;xr4;xr5];
166
167 % Calculando os erros para compor a função objetivo
168 pesos1 = diag(param.wQ);
169 SP = SPseg'*ones(1,P+1);
170 erro1 = xr - SP;
171 erro1q = zeros(1,P+1);
172 parfor jj=1:P+1
173     erro1q(jj) = erro1(:,jj)'*pesos1*erro1(:,jj);      % erro1 ao quadrado
174 end
175
176 erro1q = sum(erro1q);
177

```

```

178 delta1 = zeros(1,C-1);
179 delta2 = zeros(1,C-1);
180 delta3 = zeros(1,C-1);
181 delta4 = zeros(1,C-1);
182
183 delta1(1:C-1) = u0((1:C-1)+1)-u0(1:C-1);
184 delta2(1:C-1) = u0(C+(1:C-1)+1)-u0(C+(1:C-1));
185 delta3(1:C-1) = u0(2*C+(1:C-1)+1)-u0(2*C+(1:C-1));
186 delta4(1:C-1) = u0(3*C+(1:C-1)+1)-u0(3*C+(1:C-1));
187
188 deltaq = zeros(1,4);
189 deltaq(1) = delta1*delta1';
190 deltaq(2) = delta2*delta2';
191 deltaq(3) = delta3*delta3';
192 deltaq(4) = delta4*delta4';
193
194 pesos2 = param.wR;
195 erro2q = deltaq*pesos2;
196
197 if param.WS
198     pesos3 = param.wS;
199     pesos3 = diag(reshape((pesos3*ones(1,C))',1,[]));
200     erro3 = u0 - reshape((param.FeedEE*ones(1,C))',1,[]);
201     erro3q = erro3*pesos3*erro3';
202 else erro3q = 0;
203 end
204
205 pond1 = param.WQ;
206 pond2 = param.WR;
207 pond3 = param.WS;
208
209 FO = erro1q*pond1 + erro2q*pond2 + erro3q*pond3;

```

SIMULADOR DE FALHAS

```

1 function [simsegOUT,prisc] = simseg(kk,x0plant,SPseg,u0,param,parmod,...
2     histVM,histhA,histxrecon)
3 %===== simseg =====
4 % Este programa está simulando a planta com um problema na camisa. O
5 % controlador é idêntico ao processo normal. O objetivo é testar o
6 % desempenho do controlador frente à falha na camisa.
7 %=====
8
9 d = param.d;
10 C = param.C;
11 FA = param.FA;
12 lbrec = param.lbrec;
13 ubrec = param.ubrec;
14 opRecon = param.opRecon;
15 param.kk = kk;
16 histTc = param.histTc(1:kk); % tamanho 10 em kk = 10
17 histFc = param.histFc(1:kk); % tamanho 10 em kk = 10
18 histFin = param.histFin(1:kk);
19 histFins = param.histFins(1:kk);
20 param.simse = 1;
21
22 Ns = 100; % Número de movimentos simulados
23 param.recon = 1; % Ativador da reconciliação (1=on 0=off)
24

```

```

25 histtplant(1) = (kk-1)/FA;
26 histx0plant(1,:) = x0plant;
27 prisc = 0;
28 % Simulando falha na agitação ou na camisa, enfim algo que diminua o hA
29 for ii = kk:1:kk+Ns;
30     disp(['ii SimSeg      = ' mat2str(ii)])
31
32 %===== Reconciliação =====
33     if param.recon && (ii~=kk)
34         u0rec = histhA(ii);
35         [hAcalc] = fmincon(@(u0rec)Reconc(u0rec,ii,histxrecon,histVM,...
36                             param,paramod),u0rec,[],[],[],[],lbrec,ubrec,...
37                             [],opRecon);
38         [~,xrecon] = Reconc(hAcalc,ii,histxrecon,histVM,param,paramod);
39         paramod.hA = hAcalc;
40         x0contr = xrecon;
41         histxrecon(ii,:) = xrecon;
42         histhA(ii+1,1) = hAcalc;
43     else
44         x0contr = x0plant; % Este x0plant é, na verdade, o x0contr no main
45     end
46
47     if x0contr(3) > param.Tlim
48         prisc = 1;
49         break
50     end
51
52 %===== Controle ===== ( Este controlador atua somente na Tc )
53     [TFcalc] = Controlador(u0,ii,x0contr,0,SPseg,param,paramod);
54     Tcvec = TFcalc(1:C);
55     Fcvec = TFcalc(C+1:2*C);
56     Finvec = TFcalc(2*C+1:3*C);
57     Finsvec = TFcalc(3*C+1:4*C);
58
59 %===== Planta =====
60     tempol=((ii-1):d:(ii))/FA;
61     paramod.Tc = Tcvec(1);
62     paramod.Fincat = Fcvec(1);
63     paramod.Fin = Finvec(1);
64     paramod.Fins = Finsvec(1);
65
66     switch param.falha %<<<<<<< Acidente (mostrando a falha para o contr)
67     case {'Fincat'}
68         paramod.Fincat = paramod.Fincatfail;
69     case {'Tc'}
70         paramod.Tc = paramod.Tcfail;
71     case {'UA'}
72         if ~param.recon
73             paramod.hA = paramod.hAfail;
74         end
75     end
76     paramod.falha = param.falha;
77     disp(['Estados atuais:      ' mat2str(x0plant,4)])
78     disp(['Estados estimados: ' mat2str(histxrecon(ii,:),4)])
79     disp(['Tempo SimSeg      = ' mat2str(histtplant(end),3)])
80     disp(' ')
81
82     [tplanta xplanta]=dassl('PlantaFail',tempol,x0plant,[],paramod);

```

```

83
84 %===== Prepara para próxima iteração =====
85     u0 = [Tcvec(2:C)  Tcvec(C)  Fcvec(2:C)  Fcvec(C) ...
86           Finvec(2:C)  Finvec(C)  Finsvec(2:C)  Finsvec(C)];
87     x0plant = xplanta(end,:);
88     VM = x0plant(3) + normrnd(0,0.2);
89
90     histx0plant(ii+2-kk,:) = x0plant;
91     histtplant(ii+2-kk,1) = tplanta(end);
92     histVM(ii+1,1) = VM;
93     histTc(ii+1,1) = parmod.Tc;
94     histFc(ii+1,1) = parmod.Fincat;%<<<<<<< Acidente (mostrando a falha
95     histFin(ii+1,1) = parmod.Fin;%                                     para o reconciliador)
96     histFins(ii+1,1) = parmod.Fins;
97     param.histTc = histTc;
98     param.histFc = histFc;
99     param.histFin = histFin;
100    param.histFins = histFins;
101 end
102
103 simsegOUT.histxrecon = histxrecon;
104 simsegOUT.histhA = histhA;
105 simsegOUT.histVM = histVM;
106 simsegOUT.histTc = histTc;
107 simsegOUT.histFc = histFc;
108 simsegOUT.histFin = histFin;
109 simsegOUT.histFins = histFins;
110 simsegOUT.histx0plant = histx0plant;
111 simsegOUT.histtplant = histtplant;
112 simsegOUT.tseg = tplanta(end);

    BUSCADOR DE ESTADOS SEGUROS
1 function [SPseg,calcsegOUT] = calcseg(x0plant,ii,parmod,param,histhA)
2
3 % Simula algumas horas de operação sob refrigeração máxima do reator
4 % (full cooling) tendo por objetivo encontrar uma condição inicial que
5 % garanta que a temperatura máxima do reator não excederá um limite
6 % predeterminado
7
8 optNLP = optimset('Algorithm','interior-point','UseParallel','always',...
9                  'display','off','TolX',1e-5,'TolFun',1e-5,...
10                 'TolCon',1e-5,'MaxFunEvals',8000);
11
12 % Limites para as var manip
13 lb =[param.ContrBounds.lbTc param.ContrBounds.lbFc ...
14      param.ContrBounds.lbFin param.ContrBounds.lbFs];
15 ub =[param.ContrBounds.ubTc param.ContrBounds.ubFc ...
16      param.ContrBounds.ubFin param.ContrBounds.ubFs];
17
18 % Fazendo uma média dos últimos 5 valores de hA
19 histhA = histhA(histhA>0);
20 parmod.hA = mean(histhA);
21
22 % Construindo o vetor de chutes iniciais
23 FeedSeg0 = [parmod.Tc parmod.Fincat parmod.Fin parmod.Fins];
24
25 % Calcula os novos setpoints de segurança
26 [FeedSeg,~,CSflag] = fmincon(@(FeedSeg0)FOCalcSeg(FeedSeg0,x0plant,ii,...

```

```

27         param,parmod),FeedSeg0,[],[],[],[],lb,ub,...
28         @(FeedSeg0)RestrCalcSeg(FeedSeg0,x0plant,...
29         ii,parmod,param),optNLP);
30
31 % Mostra se o estado calculado viola o limite de segurança
32 disp('Verificando os limites de segurança')
33 [crcs,~,calcsegOUT] = RestrCalcSeg(FeedSeg,x0plant,ii,parmod,param);
34 if crcs<0, OK = 'OK'; else OK = 'Deu Ruim!'; end
35 param.FeedEE = calcsegOUT.FeedEE;
36 calcsegOUT.crcs = crcs;
37 SPseg = calcsegOUT.SPseg;
38 disp(['SPseg ' mat2str(SPseg,4) 'flag: [' mat2str(CSflag) ']' OK...
39      mat2str(crcs(crcs>=0),4)])
40
41
42 function FO = FOCalcSeg(FeedSeg,x0plant,ii,param,parmod)
43
44 % Integra o modelo sem falha com Cini = x0plant com step nas varmanip igual
45 % a FeedSeg
46
47 parmod.Tc      = FeedSeg(1);
48 parmod.Fincat  = FeedSeg(2);
49 parmod.Fin     = FeedSeg(3);
50 parmod.Fins    = FeedSeg(4);
51
52 P = param.Hcvg;
53 tempo = ((ii-1):(P+ii-1))/param.FA;
54
55 [~,xr] = dasslc('ModeloContr',tempo,x0plant,[],parmod);
56
57 erro = xr(end,:) - x0plant;
58 pesos = diag([1 0 1 1 100]);
59 FO = erro*pesos*erro';
60
61
62
63
64 function [ c, ceq, calcsegOUT ] = RestrCalcSeg(FeedSeg,x0plant,ii,parmod,param)
65
66 % Integra o modelo sem falha com Cini = x0plant com step nas varmanip igual
67 % a FeedSeg
68 parmod.Tc      = FeedSeg(1);
69 parmod.Fincat  = FeedSeg(2);
70 parmod.Fin     = FeedSeg(3);
71 parmod.Fins    = FeedSeg(4);
72
73 P = param.Hcvg;
74 tempo = ((ii-1):(P+ii-1))/param.FA;
75
76 [~,EEseg,dEEseg] = dasslc('ModeloContr',tempo,x0plant,[],parmod);
77 xini = EEseg(end,:); % Estado estacionário para a alimentação FeedSeg
78
79 % Alocando espaço na memória
80 xr3 = zeros(1,P+1);
81 xr3(1) = xini(3);
82
83 if nargout == 3
84 xr1 = zeros(1,P+1);

```



```

143 xr = [xr1;xr2;xr3;xr4;xr5]';
144 % Armazena dados para o plot
145 calcsegOUT.FeedEE = FeedSeg;
146 calcsegOUT.SPseg = EEseg(end,:);
147 calcsegOUT.csx = xr;
148 calcsegOUT.cst = tempo';
149 end
150
151
152

```

MODELO DO CONTROLADOR

```

1 function [res ires] = ModeloContr(~, x, dxdt, parmod)
2
3 Tin = parmod.Tin;
4 k0 = parmod.k0;
5 Ea = parmod.Ea;
6 RR = parmod.RR;
7 dH = parmod.dH;
8 Cp = parmod.Cp;
9
10 hA = parmod.hA;
11 Tc = parmod.Tc;
12 Fincat = parmod.Fincat;
13 Fin = parmod.Fin;
14 Fins = parmod.Fins;
15
16 ##### Variáveis de estado #####
17 Ma = x(1); % Quantidade de matéria de A no CSTR (mol)
18 Mb = x(2); % Quantidade de matéria de B no CSTR (mol)
19 T = x(3); % Temperatura no CSTR (K)
20 S = x(4); % Quantidade de matéria de S no CSTR (mol)
21 Cat = x(5);
22
23 ##### Sistema de EA's #####
24 Mt = Ma + Mb + S + Cat;
25 xMa = Ma/Mt;
26 xMb = Mb/Mt;
27 xS = S/Mt;
28 xCat = 1-xMa-xMb-xS;
29 Fliq = Fin + Fins + Fincat;
30 taxa = xMa*xCat*k0*exp(-Ea/(RR*T));
31
32 ##### Sistema de EDO's #####
33
34 res(1) = Fin - Fliq*xMa - taxa - dxdt(1);
35 res(2) = taxa - Fliq*xMb - dxdt(2);
36 res(3) = (Fin+Fins)*Tin/Mt - Fliq*T/Mt + dH*taxa/(Mt*Cp) + hA*(Tc-T) - dxdt(3);
37 res(4) = Fins - Fliq*xS - dxdt(4);
38 res(5) = Fincat - Fliq*xCat - dxdt(5);
39
40 ires = 0;
41 end
42

```

PLANTA

```

1 function [res ires] = Planta(t, x, dxdt, parmod)
2 % Modelo A + S --> B + S

```

```

3
4 Tin = parmod.Tin;
5 k0 = parmod.k0;
6 Ea = parmod.Ea;
7 RR = parmod.RR;
8 dH = parmod.dH;
9 Cp = parmod.Cp;
10
11 hA = parmod.hAplant;
12 % if t >= 1
13 %     hA = 0.1+0.9*exp(-(t-1));
14 %     hA = parmod.hAfail;
15 %     hA = 0.1;
16 % else
17 %     hA = parmod.hAplant;
18 % end
19
20 if parmod.simfalha && strcmp(parmod.falha,'UA') && t >= 4
21     hA = parmod.hAfail;
22 % hA = 0.1+0.9*exp(-(t-1));
23 elseif parmod.simfalha && strcmp(parmod.falha,'Fincat') && t >= 4
24     parmod.Fincat = parmod.Fincatfail;
25 elseif parmod.simfalha && strcmp(parmod.falha,'Tc') && t >= 4
26     parmod.Tc = parmod.Tcfail;
27 end
28
29 Tc = parmod.Tc;
30 Fincat = parmod.Fincat;
31 Fin = parmod.Fin;
32 Fins = parmod.Fins;
33
34 ##### Variáveis de estado #####
35 Ma = x(1);           % Quantidade de matéria de A no CSTR (mol)
36 Mb = x(2);           % Quantidade de matéria de B no CSTR (mol)
37 T = x(3);            % Temperatura no CSTR (K)
38 S = x(4);            % Quantidade de matéria de S no CSTR (mol)
39 Cat = x(5);
40
41 ##### Sistema de EA's #####
42 Mt = Ma + Mb + S + Cat;
43 xMa = Ma/Mt;
44 xMb = Mb/Mt;
45 xS = S/Mt;
46 xCat = 1-xMa-xMb-xS;
47 Fliq = Fin + Fins + Fincat;
48 taxa = xMa*xCat*k0*exp(-Ea/(RR*T));
49
50 ##### Sistema de EDO's #####
51
52 res(1) = Fin - Fliq*xMa - taxa - dxdt(1);
53 res(2) = taxa - Fliq*xMb - dxdt(2);
54 res(3) = (Fin+Fins)*Tin/Mt-Fliq*T/Mt+dH*taxa/(Mt*Cp)+hA*(Tc-T)-dxdt(3);
55 res(4) = Fins - Fliq*xS - dxdt(4);
56 res(5) = Fincat - Fliq*xCat - dxdt(5);
57
58 ires = 0;
59 end
60

```

```

    MODELO ESTACIONÁRIO
1  function res = ModeloEE(x,parmod)
2
3  Tin = parmod.Tin;
4  k0 = parmod.k0;
5  Ea = parmod.Ea;
6  RR = parmod.RR;
7  dH = parmod.dH;
8  Cp = parmod.Cp;
9
10 hA = parmod.hA;
11 Tc = parmod.Tc;
12 Fincat = parmod.Fincat;
13 Fin = parmod.Fin;
14 Fins = parmod.Fins;
15
16 ##### Variáveis de estado #####
17 Ma = x(1);           % Quantidade de matéria de A no CSTR (mol)
18 Mb = x(2);           % Quantidade de matéria de B no CSTR (mol)
19 T = x(3);           % Temperatura no CSTR (K)
20 S = x(4);           % Quantidade de matéria de S no CSTR (mol)
21 Cat = x(5);
22
23 ##### Sistema de EA's #####
24 Mt = Ma + Mb + S + Cat;
25 xMa = Ma/Mt;
26 xMb = Mb/Mt;
27 xS = S/Mt;
28 xCat = 1-xMa-xMb-xS;
29 Fliq = Fin + Fins + Fincat;
30 taxa = xMa*xCat*k0*exp(-Ea/(RR*T));
31
32 ##### Sistema de EDO's #####
33
34 res(1) = Fin - Fliq*xMa - taxa;
35 res(2) = taxa - Fliq*xMb;
36 res(3) = (Fin+Fins)*Tin/Mt - Fliq*T/Mt + dH*taxa/(Mt*Cp) + hA*(Tc-T);
37 res(4) = Fins - Fliq*xS;
38 res(5) = Fincat - Fliq*xCat;
39
40 end

```

```

    PLANTA COM FALHA
1  function [res ires] = PlantaFail(~,x,dxdt,parmod)
2  % Modelo      A + S --> B + S
3
4  Tin = parmod.Tin;
5  k0 = parmod.k0;
6  Ea = parmod.Ea;
7  RR = parmod.RR;
8  dH = parmod.dH;
9  Cp = parmod.Cp;
10
11 % if t<1
12 %     hA = parmod.hA;
13 %     Fincat = parmod.Fincat;
14 % hA = parmod.hAf;

```

