



**COPPE/UFRJ**

DESENVOLVIMENTO DE UM SISTEMA DE OTIMIZAÇÃO DINÂMICA EM  
TEMPO REAL

Otto Indio do Brasil Magalhães

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.

Orientador(es): Argimiro Resende Secchi

Enrique Luis Lima


Rio de Janeiro  
Setembro de 2010

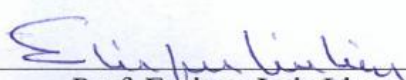
DESENVOLVIMENTO DE UM SISTEMA DE OTIMIZAÇÃO DINÂMICA EM  
TEMPO REAL

Otto Indio do Brasil Magalhães

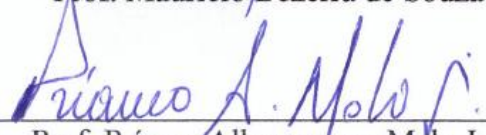
DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA  
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE  
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE  
EM CIÊNCIAS EM ENGENHARIA QUÍMICA.


Examinada por:

  
Prof. Argimiro Resende Secchi, D.Sc.

  
Prof. Enrique Luis Lima, D.Sc.

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

  
Prof. Príamo Albuquerque Melo Junior, D.Sc.

  
Dr. Marcos Vinicius de Carvalho Gomes, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2010

Magalhães, Otto Indio do Brasil

Desenvolvimento de um Sistema de Otimização Dinâmica em Tempo Real / Otto Indio do Brasil Magalhães. – Rio de Janeiro: UFRJ/COPPE, 2010.

XII, 131 p.: il.; 29,7 cm.

Orientadores: Argimiro Resende Secchi

Enrique Luis Lima

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia Química, 2010.

Referencias Bibliográficas: p. 109-123.

1. D-RTO. 2. Otimização Dinâmica. 3. Controle Ótimo. 4. Métodos Sequenciais. I. Secchi, Argimiro Resende *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

*“Buscai primeiro o reino de Deus, e a sua justiça...”*

*Mateus 6:33*

## **Agradecimentos**

Aos meus orientadores, Argimiro Resende Secchi e Enrique Luis Lima, pela oportunidade, confiança e paciência ao transmitir um pouco dos seus conhecimentos a mim.

Aos meus pais, Antônio Carlos e Iara, que sempre estiveram ao meu lado me incentivando em todos os desafios que enfrentei na minha vida.

À minha esposa Thalita, que soube entender a minha ausência e me deu o apoio necessário para que eu pudesse concluir este trabalho.

Ao meu irmão mais novo, João Pedro, que este trabalho sirva de motivação para seus estudos.

Às minhas lindas irmãs, Zélia e Paola, pela amizade e companheirismo de sempre.

À Chemtech pelo apoio e por permitir que eu pudesse chegar até aqui.

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

## DESENVOLVIMENTO DE UM SISTEMA DE OTIMIZAÇÃO DINÂMICA EM TEMPO REAL

Otto Indio do Brasil Magalhães

Setembro/2010

Orientadores: Argimiro Resende Secchi

Enrique Luis Lima

Programa: Engenharia Química

Desde a década de 1980, a indústria química tem investido na utilização de ferramentas de otimização estacionária em tempo real (RTO). Porém, esta tecnologia está limitada pelas suas características estacionárias, não sendo efetivo para otimizar o processo durante transições de operação e incapaz de mostrar o caminho ótimo entre condições estacionárias. Para cobrir este espaço, a otimização dinâmica em tempo real (D-RTO) é a tecnologia adequada para reduzir a quantidade de produtos fora de especificação e otimizar o lucro operacional diante destas perturbações. Com o objetivo de contribuir com a consolidação desta tecnologia, foi desenvolvida uma nova estrutura de D-RTO para operar em plantas de processo. Para tanto, utilizou-se um módulo de simulação dinâmica de processo, um servidor OPC (*OLE for Process Control*), para armazenar e fornecer os dados de planta, um módulo de otimização dinâmica e um módulo de controle preditivo linear (LMPC). Foi introduzido, também, um novo conceito de disparador de D-RTO. O sistema proposto foi testado em um reator CSTR com cinética de Van der Vusse explorando as oportunidades de otimização frente a perturbações comuns no processo. Com objetivo de analisar o desempenho do sistema, foram avaliados o Módulo de Otimização Dinâmica e o Módulo do Disparador D-RTO.

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

## DEVELOPMENT OF A DYNAMIC REAL-TIME OPTIMIZATION SYSTEM

Otto Indio do Brasil Magalhães

September/2010

Advisors: Argimiro Resende Secchi

Enrique Luis Lima

Department: Chemical Engineering

Since the 1980s, the chemical industry has invested in the use of real-time stationary optimization tools (RTO). However, this technology is limited by its stationary characteristics, not being effective to optimize the process during transitions of operational conditions and unable to show the optimal trajectory between stationary conditions. To cover this gap, the dynamic real-time optimization (D-RTO) is the suitable technology to reduce off-spec production and optimize the operational profit against these disturbances. In order to contribute to the consolidation of this technology, a new D-RTO structure was developed to operate in process plants. To do so, the following modules were developed: a process dynamic simulation module, an OPC (OLE for Process Control) Server, to store and provide the plant data, a dynamic optimization module and a linear model predictive control (LMPC) module. A new D-RTO trigger concept was also introduced. The proposed system was tested in a CSTR reactor with Van der Vusse reaction scheme exploiting the optimization opportunities against common process disturbances such as quality and feed flow transitions. In order to analyze the system performance, the dynamic optimization module and the D-RTO trigger were evaluated.

## Sumário

1	INTRODUÇÃO.....	1
2	REVISÃO DA LITERATURA .....	6
2.1	ANÁLISE CRÍTICA DAS ESTRATÉGIAS MPC, RTO E D-RTO .....	6
2.2	HISTÓRICO DA OTIMIZAÇÃO DINÂMICA .....	12
2.3	DIFERENTES ESTRUTURAS DO SISTEMA DE OTIMIZAÇÃO DINÂMICA.....	17
3	METODOLOGIA DO TRABALHO.....	27
3.1	OTIMIZAÇÃO DINÂMICA EM TEMPO REAL .....	27
3.1.1	FORMULAÇÃO DO PROBLEMA DE OTIMIZAÇÃO DINÂMICA.....	31
3.1.1.1	O PROBLEMA GERAL .....	32
3.1.1.2	FORMULAÇÃO DA FUNÇÃO OBJETIVO.....	34
3.1.1.3	FORMULAÇÃO DAS RESTRIÇÕES.....	40
3.1.2	SOLUÇÃO DO PROBLEMA DE OTIMIZAÇÃO DINÂMICA.....	47
3.2	ESTRUTURA PROPOSTA DO SISTEMA DE OTIMIZAÇÃO DINÂMICA EM TEMPO REAL 51	
3.2.1	MÓDULO DO SERVIDOR OPC.....	54
3.2.2	MÓDULO DE INICIALIZAÇÃO .....	55
3.2.3	MÓDULO DE SIMULAÇÃO DO PROCESSO .....	55
3.2.4	MÓDULO DE OTIMIZAÇÃO DINÂMICA.....	57
3.2.5	MÓDULO DE CONTROLE AVANÇADO .....	65
3.2.6	MÓDULO DISPARADOR DO D-RTO.....	66
4	ESTUDO DE CASO: CINÉTICA DE VAN DER VUSSE.....	69
4.1	DESCRIÇÃO DO PROCESSO.....	69
4.2	DINÂMICA DO SISTEMA .....	73
4.3	CONSTRUÇÃO DO PROBLEMA DE OTIMIZAÇÃO DINÂMICA.....	74
4.3.1	FORMULAÇÃO DO DAOP.....	75
4.3.2	SOLUÇÃO DO PROBLEMA DE OTIMIZAÇÃO DINÂMICA.....	76
4.4	APLICAÇÃO EM TEMPO REAL.....	81
5	RESULTADOS E DISCUSSÕES.....	85
5.1	AVALIAÇÃO DO MÓDULO DE OTIMIZAÇÃO DINÂMICA .....	85
5.2	AVALIAÇÃO DO MÓDULO DISPARADOR D-RTO .....	98
6	CONCLUSÕES.....	106
	REFERÊNCIAS BIBLIOGRÁFICAS.....	109
	APENDICE A - CÓDIGO EM MATLAB DA S-FUNCTION PARA O ESTUDO DE CASOS DO REATOR CSTR.....	124
	APENDICE B- CÓDIGO EM MATLAB PARA O MÓDULO OPC.....	127



## Índice de Figuras

Figura 2.1.1 Representação Esquemática da funcionalidade do Sistema RTO. Adaptado de Sequeira <i>et al.</i> (2002). .....	6
Figura 2.1.2 Hierarquia RTO. Adaptado de Sequeira <i>et al.</i> (2002). .....	9
Figura 2.3.1 – Estrutura geral do D-RTO – Módulo on-line. ....	17
Figura 2.3.2 – Estrutura geral da otimização dinâmica <i>off-line</i> . ....	19
Figura 2.3.3 – Esquema geral do D-RTO em uma camada. ....	23
Figura 2.3.4 – Estimador de estados, (a) somente D-RTO, (b) para o D-RTO e MPC/NMPC. ....	23
Figura 2.3.5 – Estimadores de estados distintos para D-RTO e MPC. ....	24
Figura 2.3.6 – Sistema D-RTO com disparador. ....	25
Figura 2.3.7 – Validação de Resultados do D-RTO. ....	25
Figura 3.1.1.1 – Formulação do Problema de Otimização Dinâmica. ....	31
Figura 3.1.2.1 – Métodos de Solução de problemas de otimização dinâmica (DAOP). ....	51
Figura 3.2.1 – Estrutura simplificada do sistema de D-RTO. OV – Variáveis de saída da planta; MD – Perturbações medidas; UD – Perturbações não medidas; MV – Variáveis manipuladas. ....	53
Figura 3.2.4.1 – Método <i>single-shooting</i> direto. ....	60
Figura 4.1.1 – Esboço de um CSTR com resfriamento e reação de Van der Vusse. ....	72
Figura 4.2.1 – $C_B$ como função de $F_f$ para $T = 378,05K$ e diferentes concentrações de $C_{Af}$ . ....	74
Figura 4.3.2.1 – Funções constantes por partes. ....	77
Figura 4.4.1 – Equações do modelo do reator CSTR com cinética de Van der Vusse na linguagem do simulador de processos EMSO. ....	82
Figura 4.4.2 – Configuração da interface EMSO-OPC. ....	83
Figura 4.4.3 – Configuração do Servidor OPC para estudo de casos do reator CSTR com reação de Van der Vusse. ....	84
Figura 5.1.1- Concentração de B sem perturbação com D-RTO desativado (MPC) e ativado. ....	87
Figura 5.1.2 – Temperatura do reator sem perturbação com D-RTO desativado (MPC) e ativado. ....	87
Figura 5.1.3- Concentração de A sem perturbação com D-RTO desativado (MPC) e ativado. ....	88
Figura 5.1.4- Concentração de C sem perturbação com D-RTO desativado (MPC) e ativado. ....	88
Figura 5.1.5- Concentração de D sem perturbação com D-RTO desativado (MPC) e ativado. ....	88
Figura 5.1.6- Vazão de alimentação sem perturbação com D-RTO desativado (MPC) e ativado. ....	88
Figura 5.1.7- Temperatura de resfriamento sem perturbação com D-RTO desativado (MPC) e ativado. ....	89
Figura 5.1.8- Concentração de B, perturbação caso 2, com D-RTO desativado (MPC) e ativado. ....	90
Figura 5.1.9 – Temperatura do reator, perturbação caso 2, com D-RTO desativado (MPC) e ativado. ....	90
Figura 5.1.10- Concentração de A, perturbação caso 2, com D-RTO desativado (MPC) e ativado. ....	90
Figura 5.1.11- Concentração de C, perturbação caso 2, com D-RTO desativado (MPC) e ativado. ....	90

Figura 5.1.12- Concentração de D, perturbação caso 2, com D-RTO desativado (MPC) e ativado.....	91
Figura 5.1.13- Vazão de alimentação, perturbação caso 2, com D-RTO desativado (MPC) e ativado.....	91
Figura 5.1.14- Temperatura de resfriamento, perturbação caso 2, com D-RTO desativado (MPC) e ativado. ....	91
Figura 5.1.15- Concentração de B, perturbação caso 3, com D-RTO desativado (MPC) e ativado.....	92
Figura 5.1.16- Temperatura do reator, perturbação caso 3, com D-RTO desativado (MPC) e ativado.....	92
Figura 5.1.17- Concentração de A, perturbação caso 3, com D-RTO desativado (MPC) e ativado.....	92
Figura 5.1.18- Concentração de C, perturbação caso 3, com D-RTO desativado (MPC) e ativado.....	92
Figura 5.1.19- Concentração de D, perturbação caso 3, com D-RTO desativado (MPC) e ativado.....	93
Figura 5.1.20- Vazão de alimentação, perturbação caso 3, com D-RTO desativado (MPC) e ativado.....	93
Figura 5.1.21- Temperatura de resfriamento, perturbação caso 3, com D-RTO desativado (MPC) e ativado. ....	93
Figura 5.1.22- Concentração de B, perturbação caso 4, com D-RTO desativado (MPC) e ativado.....	94
Figura 5.1.23- Temperatura do reator, perturbação caso 4, com D-RTO desativado (MPC) e ativado.....	94
Figura 5.1.24- Concentração de A, perturbação caso 4, com D-RTO desativado (MPC) e ativado.....	95
Figura 5.1.25- Concentração de C, perturbação caso 4, com D-RTO desativado (MPC) e ativado.....	95
Figura 5.1.26- Concentração de D, perturbação caso 4, com D-RTO desativado (MPC) e ativado.....	95
Figura 5.1.27- Vazão de alimentação, perturbação caso 4, com D-RTO desativado (MPC) e ativado.....	95
Figura 5.1.28- Temperatura de resfriamento, perturbação caso 4, com D-RTO desativado (MPC) e ativado. ....	96
Figura 5.1.29- Concentração de B, perturbação caso 5, com D-RTO desativado (MPC) e ativado.....	97
Figura 5.1.30- Temperatura do reato, perturbação caso 5, com D-RTO desativado (MPC) e ativado.....	97
Figura 5.1.31- Concentração de A, perturbação caso 5, com D-RTO desativado (MPC) e ativado.....	97
Figura 5.1.32- Concentração de C, perturbação caso 5, com D-RTO desativado (MPC) e ativado.....	97
Figura 5.1.33- Concentração de D, perturbação caso 5, com D-RTO desativado (MPC) e ativado.....	98
Figura 5.1.34- Vazão de alimentação, perturbação caso 5, com D-RTO desativado (MPC) e ativado.....	98
Figura 5.1.35- Temperatura de resfriamento, perturbação caso 5, com D-RTO desativado (MPC) e ativado. ....	98
Figura 5.2.1- Perfil da concentração de A na corrente de alimentação. ....	99
Figura 5.2.2- Concentração de B caso 1: forma cíclica. ....	101

Figura 5.2.3 – Temperatura do reator caso 1: forma cíclica.....	101
Figura 5.2.4- Concentração de A caso 1: forma cíclica. ....	102
Figura 5.2.5- Concentração de C caso 1: forma cíclica. ....	102
Figura 5.2.6- Concentração de D caso 1: forma cíclica. ....	102
Figura 5.2.7- Vazão de alimentação caso 1: forma cíclica.....	102
Figura 5.2.8- Temperatura de resfriamento caso 1: forma cíclica. ....	102
Figura 5.2.9- Concentração de B caso 2: forma inteligente. ....	103
Figura 5.2.10 – Temperatura do reator caso 2: forma inteligente. ....	103
Figura 5.2.11- Concentração de A caso 2: forma inteligente. ....	103
Figura 5.2.12- Concentração de C caso 2: forma inteligente. ....	103
Figura 5.2.13- Concentração de D caso 2: forma inteligente. ....	104
Figura 5.2.14- Vazão de alimentação caso 2: forma inteligente. ....	104
Figura 5.2.15- Temperatura de resfriamento caso 2: forma inteligente.....	104

## Índice de Tabelas

Tabela 2.3.1 – Resumo de Alguns pacotes de otimização dinâmica .....	22
Tabela 4.1.1 – Parâmetros do Modelo .....	72
Tabela 5.1.1 – Planejamento de perturbações para avaliação da camada de otimização	86
Tabela 5.2.1 – Estudo de casos com Disparador de DRTO .....	100
Tabela 5.2.2 – Resultados da avaliação do módulo Disparador de DRTO.....	100

# 1 Introdução

Com a crescente competitividade estabelecida entre as indústrias concorrentes no mercado, regida pelas quedas nas margens de lucro e pressão pela redução do tempo decorrido entre o pedido e a entrega de produtos de forma confiável, as unidades de processo precisam operar próximas dos seus limites, de forma segura. Além disso, o mercado consumidor exige cada vez mais flexibilidade por parte das unidades operacionais de forma a produzir com as características desejadas pelo cliente. Estes fatores fizeram com que a indústria quebrasse certos paradigmas operacionais. Como consequência a indústria de processamento passou a se interessar em operar as plantas de forma otimizada através de computadores, isto é, dentre as diversas formas de se formular um problema de otimização de uma unidade de processo industrial, pode-se citar uma como:

*“Operar a unidade de processo definindo as ações de controle que otimizem o seu desempenho, respeitando os critérios de segurança operacional, as especificações de qualidade dos produtos, as restrições operacionais e de mercado, e os limites de emissões ambientais.”*

Para equacionar este problema, a indústria vem aprimorando as suas ferramentas de controle e otimização.

Na década de 1960, a indústria de processamento procurou utilizar os conceitos de controle ótimo apresentados por Pontryagin (1962), porém sem muitos resultados práticos. No final da década de 1970, aproveitando os conceitos de controle ótimo da

década de 1960, surgiram as primeiras soluções de controle preditivo baseadas em modelos (MPC). Estes controladores tiveram uma grande aceitação nas indústrias de processamento, mas com uma maior difusão na década de 90. Nesta mesma época, a academia investiu esforços de seus pesquisadores em resolver os problemas de otimização estática e dinâmica. Na década de 1980, surgiram as primeiras soluções de otimização estática em tempo real (RTO), porém somente se estabeleceram na década de 1990, devido a uma série de problemas recorrentes de implementação e de manutenção de modelos do processo. As soluções de otimização dinâmica tiveram sua evolução somente no modo *off-line* em estudos de casos. Nestes casos, são estabelecidas as condições iniciais do processo e um determinado cenário a ser otimizado. As soluções de otimização dinâmica em tempo real (D-RTO) e de controle preditivo não linear (NMPC) passaram a ter um interesse efetivo pela indústria de processamento recentemente, depois que houve melhorias de robustez, desempenho e precisão dos métodos de otimização e dos algoritmos de NLP (*Non-Linear Programming*) e também no desempenho dos computadores. Até então o interesse da indústria se restringia à simulação dinâmica, mas hoje se percebe um interesse em sua otimização.

Há uma grande gama de aplicações de soluções de Simulação/Otimização Dinâmica. Podem-se destacar as seguintes:

- Determinação de condições ótimas de operação
- Análise e otimização de partidas e paradas de unidades
- Treinamento de operadores em tempo real ou acelerado
- Geração de modelos lineares e não-lineares para o controle avançado
- Calibração de modelos com dados experimentais
- Sintonia ótima de controladores PID

- Síntese e análise de estratégias de controle
- Monitoração de desempenho do processo
- Análise de sensibilidade do processo a perturbações
- Controle ótimo
- Simulação de situações anormais e procedimentos de segurança

A solução de problemas de otimização dinâmica, na maioria dos casos, passa por uma etapa de discretização de variáveis do problema e de otimização de um problema de NLP. Em função destas características, surgiu uma série de métodos de otimização dinâmica, que podem ser classificados quanto:

- **Ao uso das condições de otimalidade** – métodos indiretos, diretos e híbridos;
- **À discretização de variáveis** – discretização parcial e total;
- **À forma de obter a solução ótima** – métodos sequenciais e simultâneos;
- **À viabilidade das trajetórias** – métodos de caminho viável e inviável (*feasible e infeasible path*);
- **À obtenção das trajetórias dos estados** – *single-shooting, multiple-shooting*, colocação e colocação em elementos finitos.

Com o grau de maturidade alcançada com os MPCs e RTOs, o rigor na operação das unidades de processo vêm aumentando sistematicamente. Com isso, hoje os recursos oferecidos pelos controladores preditivos e otimizadores estáticos estão se esgotando. Diante das perturbações cada vez mais frequentes nas matérias-primas,

programações de produção e transições operacionais (principalmente em processos em batelada ou semi-batelada) fazem com que a solução de D-RTO se torne atraente por parte das indústrias de processamento. Porém, o grau de maturidade desta tecnologia ainda é mediano.

Com o objetivo de contribuir com a consolidação desta tecnologia, o presente trabalho consiste em desenvolver um Sistema de Otimização Dinâmica em Tempo Real (D-RTO), utilizando como base os softwares MATLAB/SIMULINK e EMSO (*Environment for Modeling, Simulation and Optimization*). Como objeto de pesquisa, foi desenvolvido uma estrutura de otimização dinâmica que possa operar nas plantas de processo. Realizou-se um estudo de casos aplicando o sistema de D-RTO desenvolvido a um reator CSTR com cinética de Van der Vusse onde para a solução do problema de otimização dinâmica foi implementado um método direto com *single-shooting*, que é um método de discretização parcial, sequencial e de caminho viável. Neste estudo foram exploradas as oportunidades de otimização frente a perturbações comuns no processo como mudanças de carga (qualidade e quantidade) e transições de processo.

Este trabalho contém 6 capítulos, distribuindo os assuntos na seguinte forma:

**Capítulo 2 – Revisão da Literatura.** Neste capítulo, foi feita uma revisão da literatura, iniciando com as estruturas de RTO bem como suas principais vantagens e desvantagens; fez-se uma análise crítica da otimização dinâmica frente às soluções de MPC, RTO e D-RTO. Além disso, foi feito um resumo da evolução da otimização dinâmica, passando pelos históricos do cálculo variacional, do controle ótimo, da teoria de controle moderno, do controle preditivo e da otimização dinâmica. São também apresentadas as diferentes opções de estruturas de sistemas de D-RTO.



**Capítulo 3 – Metodologia do Trabalho.** Neste capítulo, é feito um descritivo da otimização dinâmica em tempo real: as partes da formulação do problema de otimização dinâmica; as diferentes abordagens de solução de problemas de otimização dinâmica. Ao final, é apresentada a estrutura proposta de Otimização Dinâmica em Tempo Real bem como a descrição das características de cada um de seus módulos.

**Capítulo 4 – Estudo de Casos.** Neste capítulo, é apresentado o estudo da otimização dinâmica do modelo do reator CSTR com cinética de Van der Vusse utilizando o sistema de D-RTO desenvolvido neste trabalho. É apresentado o modelo do reator, a construção do problema de otimização dinâmica, a descrição do algoritmo de solução implementado e como é realizada a aplicação em tempo real.

**Capítulo 5 – Resultados e Discussões.** Neste capítulo é apresentado e discutido os resultados obtidos na aplicação do sistema de otimização dinâmica ao estudo de casos do reator CSTR com cinética de Van der Vusse; foi realizado um planejamento de perturbações partindo das condições ótimas (*set-points*) e a influência do desempenho da estrutura com a presença dos módulos da otimização dinâmica e do disparador do D-RTO foi avaliada.

**Capítulo 6 – Conclusões.** Neste capítulo, são apresentadas as conclusões do trabalho e as sugestões para trabalhos futuros.

## 2 Revisão da Literatura

### 2.1 Análise Crítica das Estratégias MPC, RTO e D-RTO

O objetivo principal de um sistema de Otimização Estacionária em Tempo Real (RTO) é operar uma planta, em cada instante de tempo, tão próximo de suas condições operacionais ótimas quanto possível. Para atingir este objetivo, um sistema de RTO contém, basicamente, os componentes descritos na Figura 2.1.1.

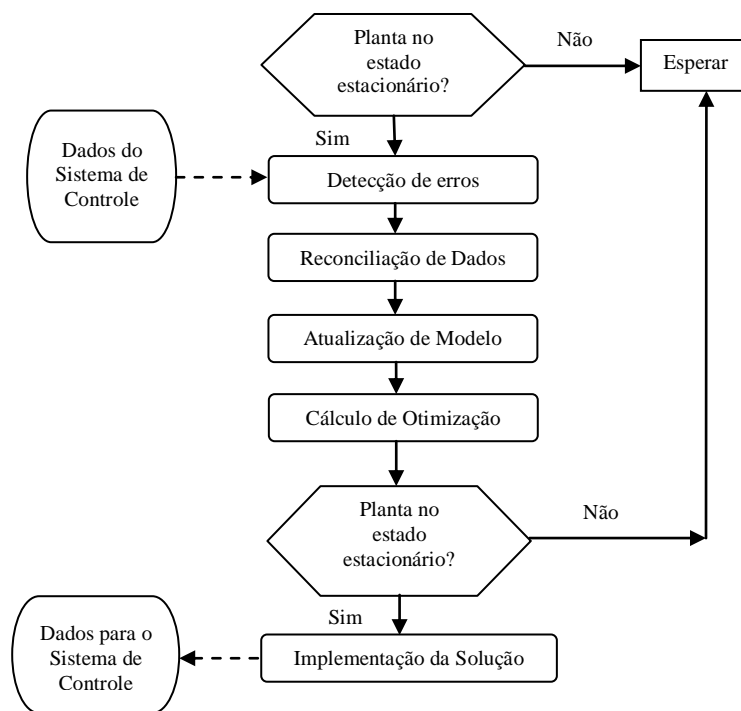


Figura 2.1.1 Representação Esquemática da funcionalidade do Sistema RTO.

Adaptado de Sequeira *et al.* (2002).

A estrutura RTO pode ser considerada como uma extensão de um sistema de controle *feedback*. Consiste de subsistemas para validação de medida, detecção de estado estacionário, atualização do modelo de processo, otimização baseada em modelo e condicionamento de comando. Uma vez que a operação da planta tenha alcançado o

estado estacionário, dados da planta são coletados e validados para evitar erros grosseiros nas medidas de processo e as medidas podem ser reconciliadas usando balanços de massa e energia para garantir a consistência do conjunto de dados usados para a atualização do modelo. Uma vez validadas, as medidas são usadas para estimar os parâmetros do modelo para garantir que o modelo represente corretamente a planta no ponto de operação corrente. Em seguida, os *set-points* ótimos do controlador são calculados usando o modelo atualizado e são transferidos para o sistema de controle após uma verificação feita pelo subsistema de condicionamento de comando.

Vamos analisar, com mais detalhe e a partir do ponto de vista da tomada de decisão, as desvantagens do sistema de RTO. Primeiro, o estado estacionário deve ser alcançado para a otimização ser realizada; isto significa que nada pode ser feito até esta condição ser satisfeita. Quando o período das perturbações é maior do que as constantes de tempo do processo, o cenário RTO não pode ser contemplado (White, 1998). Além disso, quando várias variáveis de decisão estão envolvidas, o procedimento de otimização pode requerer muito tempo. Ainda pior, se o *solver* não for suficientemente robusto, a solução pode ainda não convergir. Também, para implementar os *set-points* resultantes a partir da otimização, a planta deve ainda estar no estado estacionário original. Finalmente, a magnitude das mudanças nos *set-points* deve ser restringida por razões de segurança. Deste modo, não é claro quais *set-points* deveriam ser implementados quando mudanças em seus valores resultantes da otimização excedem os limites.

Alguns esforços têm sido feitos para evitar esses defeitos do RTO. Besl *et al.* (1998) implementaram um sistema que não espera pelo alcance do estado estacionário, mas é otimizado periodicamente, realizando a reconciliação de dados somente quando o

estado estacionário é alcançado. Infelizmente, esta abordagem nem sempre pode ser implementada porque a qualidade da otimização depende fortemente da qualidade dos dados e do modelo.

Considerando que os dados e o modelo são ajustados somente no estado estacionário, esta abordagem poderia levar à otimização de um modelo não adequado com dados não adequados, o qual poderia resultar em agressiva e inútil mudança nos valores dos *set-points*.

Adicionalmente, Cheng e Zafiriou (2000) apresentaram uma abordagem interessante, na qual a atualização do modelo é realizada implicitamente à otimização e o algoritmo de otimização é aplicado sobre a planta em passos sucessivos. Esta abordagem tem algumas desvantagens também. Primeiro, o modelo atualizado não está disponível para estudos *off-line*. Segundo, muitas mudanças de *set-points* são requeridas (sobre a planta) para avaliação do gradiente, o que pode levar para um comportamento da planta não desejado. Além disso, após cada mudança de *set-point*, o estado estacionário tem que ser alcançado para permitir uma aproximação de gradiente aceitável. Finalmente, em seu trabalho, a metodologia foi testada somente em um modelo em estado estacionário, o qual é uma abordagem muito restrita.

Uma contribuição bastante interessante veio a partir de pesquisa de controle preditivo multivariável (MPC), ao incorporar objetivos econômicos dentro do controlador para alcançar uma transição adequada entre o ponto de operação corrente e o desejado. Uma forma de incorporar economia dentro do controlador é adicionar um objetivo econômico simplificado ao objetivo quadrático padrão, conforme proposto por muitos autores (Tvrzská e Odloak, 1998; Becerra *et al.*, 1998; Nath e Alzein, 2000).

Entretanto, o fato de todo requisito de controle/otimização ser traduzido em um índice de desempenho escalar simples revela uma permuta entre controle e otimização, tornando a escolha de pesos relativos muito difícil. Uma outra alternativa, muito usada na indústria (Sorensen e Cutler, 1998; Rao e Rawlings, 1999; Qin e Badgwell, 1997; Ying e Joseph, 1999), é enviar os resultados do RTO a um otimizador em estado estacionário local de programação linear (LP) ou programação quadrática (QP) acoplado a um MPC [LP (QP)-MPC]. Tal esquema de controle cascata computa continuamente e atualiza os *set-points* usados pelo algoritmo do MPC de mais baixo nível, produzindo uma transição evolucionária com bom desempenho de controle (Figura 2.1.2).

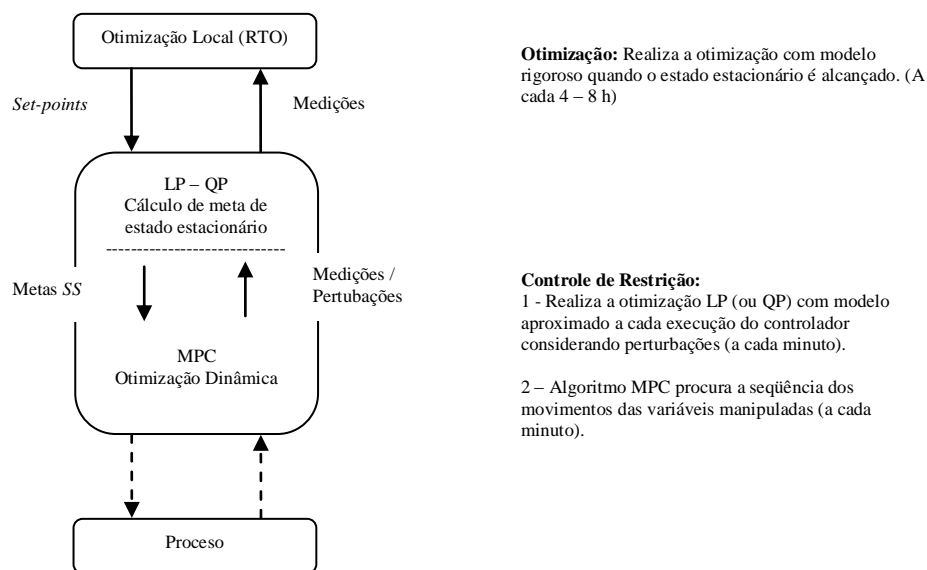


Figura 2.1.2 Hierarquia RTO. Adaptado de Sequeira *et al.* (2002).

Entretanto, em adição à complexidade de tais sistemas, problemas significativos ocorrem quando ambas as camadas do LP(QP)-MPC e RTO têm objetivos de desempenho e econômicos, os quais não podem concorrer (Mizoguchi *et al.*, 1995). Adicionalmente, também deveria ser considerado que, para certos processos, a

implementação de MPC pode não ser justificada ou pode ainda não ser uma escolha apropriada (Qin e Badgwell, 1997).

Como uma consequência da desvantagem do RTO de trabalhar particularmente com longos períodos de amostragem, muitos autores têm proposto métodos que funcionam com tempos de amostragem menores na camada de otimização. Por exemplo, Sequeira *et al.* (2002) propuseram alterar os *set-points* para a camada de controle em intervalos bem menores (no estudo de caso apresentado, 1/50 do tempo de amostragem da planta) e realizar uma evolução em tempo real (*real-time evolution*) dos *set-points* por buscas heurísticas (usadas aqui para reduzir o tempo de computação) baseadas em modelo de processo estacionário e nas medidas disponíveis. Para evitar o comportamento de *overshooting*, os passos das variáveis de decisão são restringidos. No exemplo mostrado, este método melhora o RTO de estado estacionário com o controle regulatório, especialmente para perturbações não estacionárias e na primeira fase após uma perturbação ocorrer a qual não é tão surpreendente. A idéia de que um “passo na direção correta” deveria ser melhor do que esperar até o processo assentar em um novo estado estacionário é convincente, entretanto esta abordagem sofre por negligenciar os aspectos dinâmicos, isto é, a interação da mudança de *set-point* com a camada de controle regulatório e a suposição de que uma otimização em estado estacionário realizada em um ponto de operação não estacionário proporcionaria o movimento correto dos *set-points*.

Recentemente, nesta mesma linha de pensamento, começaram as primeiras iniciativas no intuito de criar uma estrutura de D-RTO (*Dynamic Real-Time Optimization*) tal qual a estrutura de RTO (*Stationary Real-Time Optimization*). Backx, Bosgra e Marquardt (2000) propuseram uma estrutura de D-RTO integrado com

NMPC. Kadam *et al.* (2001) fizeram uma análise das diferentes estruturas de D-RTO, introduzindo o conceito de disparador de D-RTO. Almeida e Secchi (2006) discutiram uma aplicação de otimização dinâmica em tempo real para uma complexa unidade industrial de FCC (*Fluidized-Bed Catalytic Cracking*). É importante ressaltar que os sistemas de otimização dinâmica *off-line*, como o gPROMS (2001), estão consolidados e maduros, porém as aplicações em tempo real ainda estão em fase de desenvolvimento e ainda há grandes evoluções a serem feitas nesta área.

A otimização dinâmica, até a década de 1960, não tinha muito interesse por parte da indústria devido às grandes limitações da tecnologia e às dificuldades de implementação. Com a apresentação da teoria de controle ótimo e do filtro de Kalman, a otimização dinâmica teve um grande impulso, dando origem aos controladores preditivos (MPCs) usados até hoje. A partir da década de 1980, a tecnologia de otimização dinâmica tem tido um grande interesse por parte da academia e da indústria, mas utilizando-a somente em estudos de casos *off-line*.

Os resultados apresentados estimularam o desenvolvimento nesta área, permitindo que se pensasse em executá-lo *on-line*, em tempo real. Esta iniciativa reacendeu o interesse pelo desenvolvimento de NMPC, que atualmente está começando a ter aplicações industriais consolidadas. No entanto, para o D-RTO, que tem os mesmos fundamentos do NMPC, ainda há desafios e definições importantes a serem resolvidos. Dentre os pontos, pode-se considerar o critério de escolha da abordagem adequada (quando se deve escolher a abordagem variacional, sequencial, simultânea ou híbrida?), a forma de discretizar o DAOP (*Differential-Algebraic Optimization Problem*) tal que o resultado seja o mais preciso possível. Além disso, há questões ligadas à implementação em tempo real, como a inicialização do DAOP, robustez e

desempenho do algoritmo de otimização e os cuidados na implementação das ações de controle. Estes fatos fazem da solução destes problemas um desafio muito atraente e, ao resolvê-los, será alcançado um grau de maturidade muito importante para a consolidação desta tecnologia.

## 2.2 Histórico da Otimização Dinâmica

A otimização dinâmica surgiu como uma derivação do controle ótimo, pois o mesmo era resolvido através da utilização da abordagem variacional. Esta abordagem resultava em um problema de valor de contorno de difícil solução. Por este motivo, a maioria das aplicações era voltada a problemas de pequenas dimensões não sendo adequadas às indústrias de processamento, pois os problemas envolviam modelos mais complexos e um número de variáveis muito maiores.

As primeiras iniciativas mais significativas com o intuito de procurar formas para simplificar as soluções dos problemas surgiram na década de 70 com Pollard e Sargent (1970) e Sargent e Sullivan (1977). Eles utilizaram técnicas de discretização dos perfis de controle. Primeiramente eles utilizaram perfis constantes por partes para integrar os modelos em ODE (*Ordinary Differential Equations*) e resolver o problema de otimização sequencialmente. Eles foram os primeiros a produzir pacotes para resolver problemas de larga escala usando parametrização dos controles e a avaliação dos gradientes usando as equações adjuntas para poder resolver o problema de otimização resultante. Esta abordagem era do tipo de caminho viável (*feasible path*), pois produzia trajetórias viáveis para os estados a partir de perfis de controle sugeridos.



Newman e Sen (1974) utilizaram o método de Galerkin para a parametrização das variáveis de controle e de estado transformando o conjunto de equações diferenciais em equações algébricas. A primeira solução de DAOP foi apresentada por Tsang *et al.* em 1975, porém esta abordagem não teve muito uso na época.

Na década de 1980, Bock (1983) e Biegler (1984) reutilizaram a colocação ortogonal aplicada na parametrização de variáveis. Renfro (1986) e Renfro *et al.* (1987) usaram colocação *spline* global para converter o sistema DAE com os perfis de controle constantes por partes. Em 1987, Cuthrell e Biegler já notaram os problemas de precisão e estabilidade no uso de colocação global propondo assim o uso de colocação em elementos finitos para aproximar os estados e controles usando polinômios de Lagrange para converter os sistemas DAEs em equações algébricas. Em 1989, Logsdon e Biegler mostraram que o uso do método de colocação em elementos finitos era equivalente a usar o método de Runge-Kutta totalmente implícito, mas de passo constante. Por esta razão, consideraram que as suas propriedades de estabilidade e erros eram satisfatórias para uma discretização precisa de modelos DAOP. Esta representação oferece menores números de condicionamento e menores erros de arredondamento.

Cuthrell e Biegler (1989) aplicaram o método em uma série de processos químicos. O método permitiu tratar discontinuidades nos controles e em alguns estados. Com isso, a abordagem simultânea se consolidou como uma boa opção para solução de problemas de otimização dinâmica, pois evitava as dificuldades da parametrização do controle e estado encontradas nos outros métodos, resolvendo o problema de otimização de forma simultânea via NLP. Estes métodos também foram chamados de *infeasible path*, pois a solução do problema de otimização e a satisfação das restrições do modelo e do processo são obtidas simultaneamente. Trajetórias de controle se tornaram parte

das variáveis de otimização e as instabilidades e não-linearidades dos modelos dinâmicos puderam ser mais bem controladas, iniciando assim a resolução de DAOPs de dimensões maiores.

Em 1989, Cuthrell e Biegler, começaram a abordar sistemas rígidos (*stiff*) e problemas de NLP esparsos e de dimensões muito elevadas, forçando o desenvolvimento de algoritmos especializados para resolver estes tipos de problemas. Em 1983 surgiu uma aplicação pioneira (Locke *et al.*, 1983) onde se utilizou o método SQP parcialmente reduzido (PRSQP) muito usado para resolver DAOP de engenharia (Vasantharajan e Biegler, 1988; Vasantharajan e Biegler, 1990; Biegler *et al.*, 1995). Esta teoria só foi formalizada bem depois por Schulz (1996), que mostrou que o PRSQP é uma generalização de ambos *full-space* SQP e *reduced-space* SQP (rSQP).

Os métodos sequenciais tiveram uma retomada após a década de 90 com o avanço no desempenho dos computadores, o que resultou em muitas melhorias nesses métodos. Vassiliadis, em 1993, utilizou perfis constantes por partes e lineares por partes para a discretização parcial das variáveis de controle com polinômios de Lagrange, tendo uma série de relatos da utilização de discretização dos controles logo em seguida (Vassiliadis *et al.*, 1994a e b; Pytlak e Vinter, 1996a e b; Feehery e Barton, 1998). Em 1994, Barton e Pantelides resolverem problemas DAOP utilizando o gOPT/gPROMS desenvolvido pela mesma equipe na Inglaterra. Em 1996, Leineweber implementou uma estratégia *reduced-space* de solução de NLP de grande porte no pacote MUSCOD-II. Em 1997, Leineweber *et al.* rerepresentaram o método do *multiple-shooting* na sua versão simultânea, inspirado no método já apresentado por Plitt (1988). Esta abordagem permite o uso de valores inconsistentes dos estados através de uma formulação de DAE

relaxada proposta por Bock *et al.* (1988). Estratégia simultânea do *multiple-shooting* também foi utilizada por Tanartkit e Biegler (1995 e 1996).

Técnicas de controle ótimo foram resumidas na literatura, divididas em grupos, por Vassiliadis, em 1993. Os **métodos de discretização dos controles** (discretização parcial) são aqueles onde se estima o perfil de controle e se calcula o perfil de estado, em seguida a função objetivo e os gradientes até convergir o problema de otimização. Os gradientes podem ser calculados por perturbação ou pelas equações de sensibilidade (Vassiliadis, 1993) ou pelas equações adjuntas (Pytlak e Vinter, 1996). Esta abordagem é conhecida como método sequencial e pertence à classe dos métodos *feasible path*. O segundo grupo é o **método de programação dinâmica**, inspirado em Bellman (1957) e que foi rerepresentado por Luus (1990) na forma de IDP – *Iterative Dynamic Programming*. Houve algumas aplicações mostradas por Luus (1990) e com restrições nos estados por Luus e Rosen (1991) usando inicialmente perfis constantes por partes. Esta abordagem acabou não sendo muito utilizada. O terceiro grupo, os **métodos simultâneos** (discretização total), envolve os métodos de colocação, colocação em elementos finitos e *multiple-shooting* direto (apesar deste último estar em uma categoria intermediária entre os métodos sequenciais e os métodos simultâneos), revisitados por Betts (2001) e Biegler *et al* (2002).

Desde a década de 1980 houve uma preocupação com a precisão da discretização, tendo como iniciativas os trabalhos de Logsdon e Biegler (1989) e Schulz (1996), que utilizaram a discretização adaptativa usada para controlar o erro da discretização. Binder *et al.* (1997, 1999) propuseram uma estratégia de adaptação de malhas baseada no uso de *wavelets*, onde a estrutura da malha discreta é refinada até obter a melhor precisão possível.

Algumas otimizações dinâmicas em tempo real foram aplicadas a NMPCs usando a abordagem seqüencial, sendo chamadas de múltiplos passos (Li e Biegler, 1988). Em 2000, Biegler propôs o uso do método de colocação em NMPCs, não tendo muita aceitação.

No início da década de 1990 a solução de DAOPs começou a ter um certo grau de maturidade, e surgiu uma série de estudos práticos. Devido ao interesse da indústria por simulação dinâmica, vários pesquisadores iniciaram os trabalhos no desenvolvimento de tais ferramentas. Dentre elas pode-se citar uma série de aplicações de otimização *off-line* de DAE (Bock *et al.*, 1999; Tanartkit e Biegler, 1995 e 1996; Leineweber, 1996 e 1999; Petzold *et al.*, 1997; Pantelides *et al.* 1994; Vassiliadis *et al.* 1994). Destacam-se o trabalho de Cervantes *et al.* (2000a), que fizeram um estudo de otimização dinâmica *off-line* de uma planta de polietileno de baixa densidade, analisando transições na produção e aplicando discretização total e os trabalhos de Downs e Vogel (1993) e de Jockenhövel *et al.* (2003), que resolveram o problema da Tennessee Eastman, mas não em tempo real. Em 1995 foi usado o método de *multiple-shooting* em problemas de RTO e NMPC (Santos *et al.*, 2000). Leineweber *et al.* (1997) propuseram um esquema de reotimização rápida de processos em batelada com grandes perturbações.

Na década de 2000, a equipe de Biegler investiu um grande esforço na melhoria de algoritmos de otimização de NLP de grande escala. As primeiras iniciativas começaram em 1998, com Cervantes e Biegler, onde desenvolveram o rSQP e as primeiras iniciativas do IPOPT (*Interior Point Optimizer*). Em 2003, Bartlett e Biegler apresentaram o rSQP++, estruturando melhor o software e reformulando-o utilizando o conceito de orientação a objetos. Simultaneamente, Wächter (2002) desenvolveu um

novo algoritmo IPOPT, resolvendo o problema de otimização com restrições com o uso de funções barreira, usando filtro na etapa *linsearch* e PCG (gradiente conjugado condicionado) para resolver o problema com a Hessiana reduzida. Em 2005, Wächter e Biegler implementaram definitivamente estas facilidades no algoritmo IPOPT.

## 2.3 Diferentes Estruturas do Sistema de Otimização Dinâmica

A otimização dinâmica em tempo real (D-RTO) é uma aplicação *on-line*, sendo executada de forma cíclica e conectada à planta, onde se coletam informações de instrumentos da planta e se realiza o cálculo da trajetória ótima das variáveis de controle, com o auxílio de algoritmos de otimização dinâmica. Os valores das ações de controle calculadas, por sua vez, são implementados na planta, alterando o ponto de operação. Esta mesma aplicação também pode ser executada de forma *off-line*.

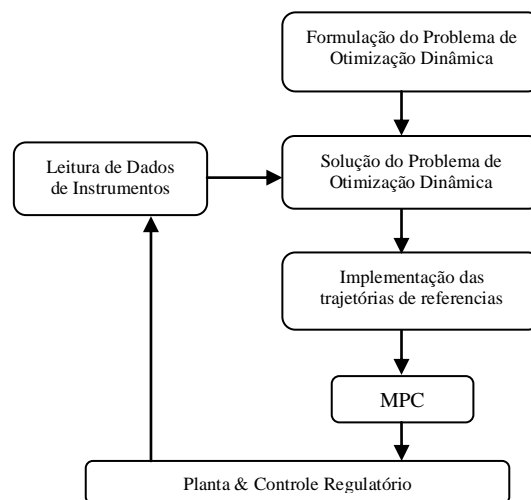


Figura 2.3.1 – Estrutura geral do D-RTO – Módulo on-line.

De forma semelhante às aplicações de otimização estacionária em tempo real (RTO – *Real Time Optimization*), para se realizar uma otimização dinâmica em tempo real, precisa-se formular o problema de otimização dinâmica (DAOP), fazer a leitura das informações da planta, inicializar a otimização, resolver o DAOP e implementar as ações de controle na planta ou em aplicações de controles avançados. Isto pode ser visto na Figura 2.3.1.

Na etapa de leitura de instrumentos e inicialização, realiza-se a coleta de dados dos instrumentos da planta, validam-se os dados coletados com eliminação de erros grosseiros, ajustam-se os parâmetros do modelo e estimam-se os estados iniciais do processo, passando previamente por uma reconciliação de dados. Na etapa de formulação do problema de otimização dinâmica, configura-se o problema de otimização em alguma interface adequada onde se estabelece a função objetivo, constrói-se o modelo do processo e se estabelecem as restrições do problema. Na etapa de solução do problema de otimização dinâmica, executa-se o tratamento do problema de otimização (ex.: discretização das variáveis). Uma vez que o problema esteja apresentado na forma adequada, um algoritmo de otimização resolve o problema de otimização dinâmica, calculando a trajetória ótima das variáveis de controle. Uma vez resolvido o problema, as trajetórias calculadas devem ser validadas e implementadas nas aplicações de MPC, na forma de trajetórias de referência para as variáveis de controle e valores ideais para as variáveis manipuladas, ou diretamente nos instrumentos da planta, na forma de *set-points* de controle regulatório.

A estrutura da otimização dinâmica *off-line* é semelhante à aplicação *on-line*, onde a leitura de dados de instrumentos da planta é substituída por um arquivo de dados e a implementação das ações de controle é substituída por um arquivo de relatório ou

um visualizador de resultados. Ao contrário da aplicação *on-line*, a aplicação *off-line* não é necessariamente executada de forma cíclica, podendo ser feita através de uma interface amigável. A Figura 2.3.2 mostra a sua estrutura geral.

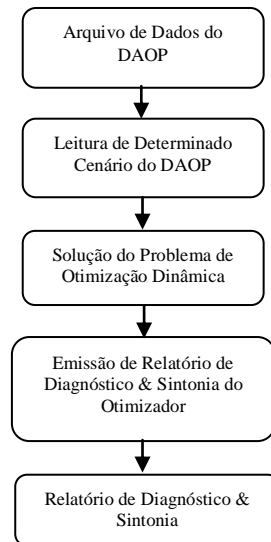


Figura 2.3.2 – Estrutura geral da otimização dinâmica *off-line*.

Devido às funções desejadas e a infra-estrutura disponível na planta, há algumas variações de estrutura de D-RTO em alguns aspectos específicos dos módulos do sistema ou na arquitetura do controle. As diferentes estratégias têm os seguintes módulos:

**Separador de Escalas de Tempo** – consiste de um módulo que separa as escalas de tempo com variações mais lentas, compatíveis com o horizonte de tempo do D-RTO, ou com perturbações de constantes de tempo menor que o horizonte de predição do MPC (Kadam *et al.*, 2001). Desta forma, definem-se os tempos de amostragem para o ciclo do D-RTO e para o ciclo do MPC.

**Estimador de Estados** – consiste de um módulo que realiza a aquisição e validação dos dados dos instrumentos da planta. Uma vez coletados os dados, realiza-se sua reconciliação dinâmica e a estimação dos estados. A reconciliação dinâmica, diferentemente do caso estacionário, não necessita que se aguarde a planta atingir o estado estacionário, podendo ser realizada a cada ciclo de amostragem de dados da planta. Como opções importantes, têm-se a utilização de filtro de Kalman estendido (EKF – *Extended Kalman Filter*) ou o estimador de horizonte móvel (MHE – *Moving Horizon Estimator*) considerado por Rao *et al.* (2001 e 2003) e Abu-el-zeet *et al.* (2002). O produto deste módulo é o fornecimento das condições iniciais consistentes para o D-RTO e MPC. O estimador pode ser único, servindo para os propósitos de D-RTO e MPC. Neste caso, deve-se reamostrar de acordo com o ciclo de execução de cada aplicação. Também há a opção de primeiramente separar as escalas de tempo e depois estimar os estados. Neste caso, deve-se estimar o estado na escala de tempo do D-RTO (escala larga), podendo utilizar modelos complexo, ou do MPC (escala curta), com o uso de modelos mais simples. Neste mesmo módulo, pode-se realizar o ajuste de parâmetros utilizando os estados estimados. O ajuste de parâmetros pode ser realizado juntamente com a estimação de estados ou em separado (Salau, 2009). O modelo a ser utilizado pode variar conforme a aplicação de MPC ou D-RTO. Neste caso, pode-se utilizar o mesmo algoritmo de otimização dinâmica, trocando-se a função objetivo para a máxima verossimilhança, ou até mesmo utilizando EKF. Para aplicações de MPC, pode-se realizar a atualização do modelo da mesma forma descrita anteriormente, porém podendo usar modelos mais simplificados.

**Disparador do D-RTO** – consiste de um módulo que dispara a execução do D-RTO. É baseado em critérios para a ativação do módulo da otimização dinâmica. Seu



uso melhora o desempenho da estrutura, pois é capaz de perceber os melhores momentos para realizar a otimização evitando execuções desnecessárias do D-RTO. Em alguns casos, a execução do D-RTO é feita com base na análise da sensibilidade do sistema às novas entradas do processo. Com base nesta sensibilidade, atualiza-se a predição e os multiplicadores de Lagrange de forma rápida e verifica-se se as variações relativas da função de Lagrange e das restrições de desigualdade foram maiores do que uma tolerância pré-estabelecida. Caso ambos os critérios sejam ultrapassados, dispara-se a execução do D-RTO, caso contrário deve-se apenas executar o MPC.

**D-RTO** – consiste de um módulo de otimização dinâmica. Este é o coração do sistema e objeto de maiores quantidades de estudos e desafios na área de D-RTO. Há diferentes abordagens utilizadas para resolver este problema, podendo ser método indireto, híbrido, *single-shooting*, *multiple-shooting*, programação dinâmica, colocação em elementos finitos, dentre outras. A grande maioria executa adaptação da discretização de malhas de tempo. Há duas abordagens básicas, a adaptação em separado ou juntamente com a solução do problema de otimização dinâmica. Nestes pacotes, há cinco blocos fundamentais, o primeiro consiste da formulação do DAOP e construção do modelo; o segundo consiste da adaptação de malhas de discretização; o terceiro consiste da discretização propriamente dita; o quarto consiste da integração do modelo e da matriz de sensibilidade (quando forem utilizados métodos sequenciais); e o quinto consiste da solução do problema de otimização através da utilização de um *solver* de NLP. As matrizes Jacobianas normalmente são obtidas com a utilização de pacotes de diferenciação automática como o ADOL-C (2004), e as matrizes Hessianas, quando necessário, podem ser obtidas por diferenciação simbólica ou numérica (por

perturbação), ou por aproximações do tipo BFGS. A Tabela 2.3.1 mostra um pequeno resumo dos sistemas mais populares.

Tabela 2.3.1 – Resumo de Alguns pacotes de otimização dinâmica.

Pacote	Grupo	Abordagem	Solver NLP
DynoPC	L.T.Biegler – CMU (Lang, 2005)	Método Simultâneo	IPOPT, rSQP++
DyOS ADOPT	W.Marquardt – RWTH (Brendel, 2004 e Schlegel, 2000)	Método Seqüencial – <i>Single Shooting</i>	SQP
gPROMS	C.C.Pantelides – PSE (gPROMS, 2001)	Método Seqüencial – <i>Single Shooting</i> , <i>Multiple shooting</i>	HQP
DYNOPT	M.Fikar - FCFT-STU Čížniar, 2006	Método Simultâneo	MATLAB
COOPT	L.R.Petzold – UCSB (Serban and Petzold, 2001)	Método Seqüencial – <i>Single Shooting</i> , <i>Multiple shooting</i>	SNOPT
CAMTOS	Klaus H.Well – IFR-SU (Gath, 2002)	Método Híbrido	SNOPT

**MPC** – consiste de um módulo de controle preditivo linear (LMPC) ou não-linear (NMPC), onde existem muitas abordagens e propostas na literatura (Rotava, 1997; Longhi, 2001).

**Validação e Implementação das Ações de Controle** – consiste de um módulo que analisa a coerência e a significância dos resultados do otimizador dinâmico. Neste módulo, realiza-se a análise dos resultados baseada em testes estatísticos.

Estes blocos podem ser usados de diferentes formas e com graus de complexidades diferentes. Os primeiros arranjos consistem em realizar a otimização dinâmica em uma camada, onde controle e otimização são realizados juntos, ou em duas

camadas (Kadam e Marquardt, 2004), conforme a Figura 2.2.3 e a Figura 2.3.4b. Estes arranjos são semelhantes ao utilizado em RTO (Zanin, 2001).

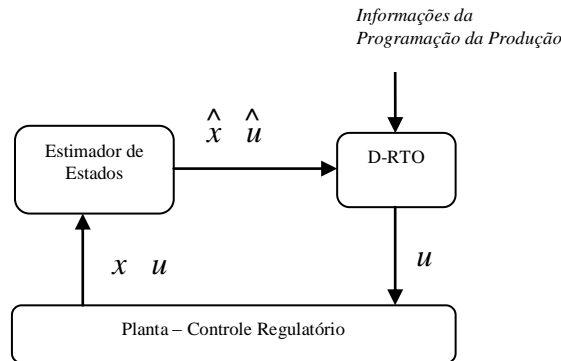


Figura 2.3.3 – Esquema geral do D-RTO em uma camada.

Em um segundo grau de complexidade, pode-se ter estimação dos estados para o D-RTO e MPC ou somente para o D-RTO (Backx *et al*, 2000). No caso de um estimador para o D-RTO e o MPC, é necessário colocar um separador de escalas de tempo (vide Figura 2.3.4).

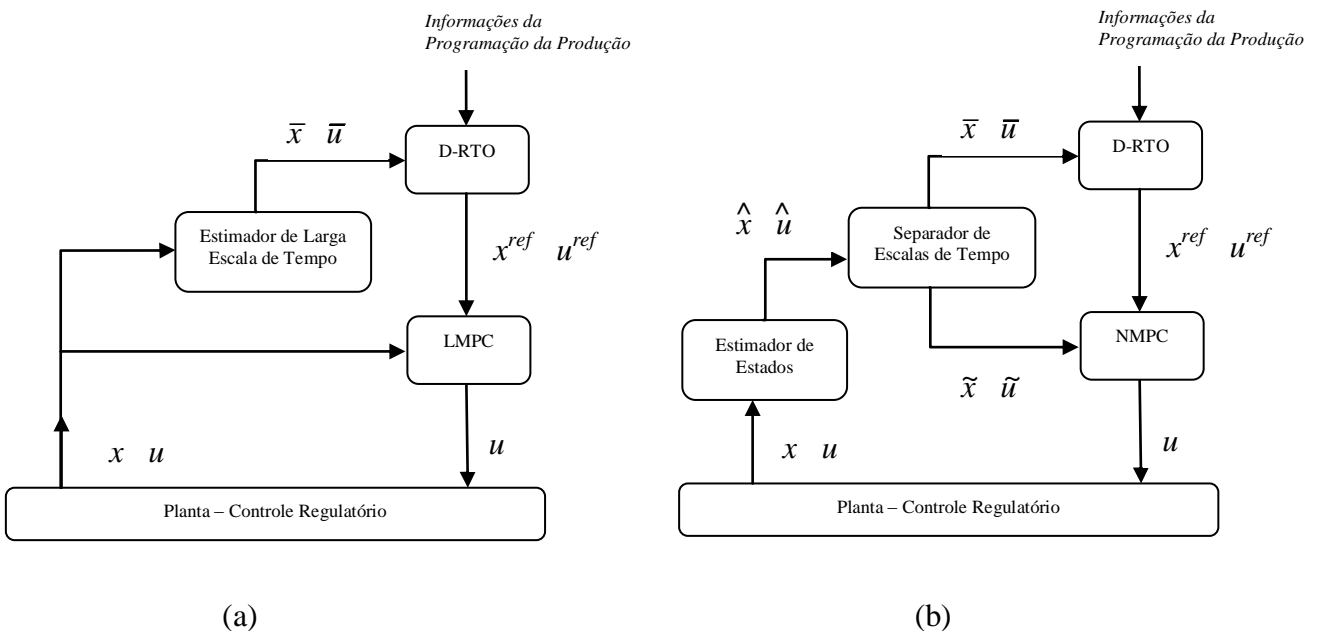


Figura 2.3.4 – Estimador de estados, (a) somente D-RTO, (b) para o D-RTO e MPC/NMPC.

O estimador de estados pode ser único para ambas as aplicações (Figura 2.3.4b, Kadam *et al*, 2001) ou diferentes (Backx *et al*, 2000). Neste ultimo caso, o estimador do D-RTO pode ter um modelo mais complexo (rigoroso) do que o modelo do MPC. Quando se utilizam estimadores distintos para as duas aplicações, o separador de escalas de tempo deve obrigatoriamente estar antes dos estimadores de estados, conforme mostra a Figura 2.3.5.

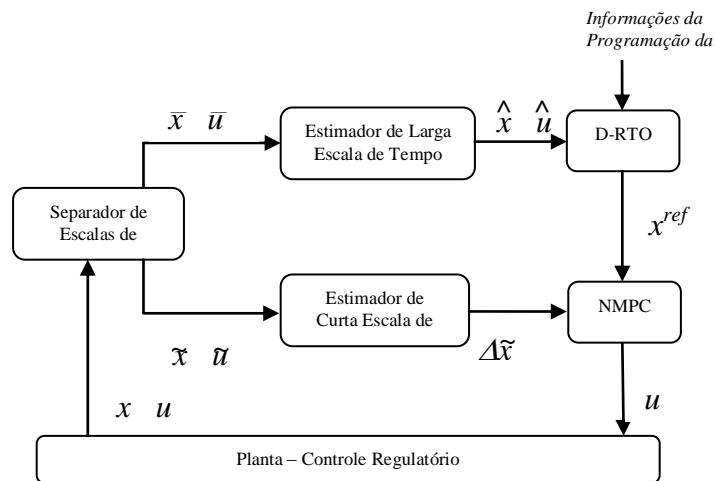


Figura 2.3.5 – Estimadores de estados distintos para D-RTO e MPC.

O sistema de D-RTO pode ser executado em um ciclo de execução pré-determinado ou ser executado somente quando há perturbação sensível no processo. Neste caso, é necessário utilizar um disparador de D-RTO (Kadam *et al*, 2003) para definir quando o D-RTO deve ser executado (vide Figura 2.3.6).

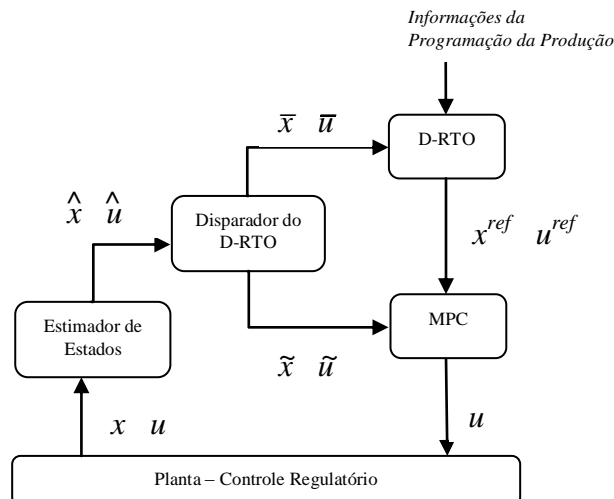


Figura 2.3.6 – Sistema D-RTO com disparador.

Finalmente, o resultado do D-RTO pode passar por uma validação final das ações de controle. As ações de controle do D-RTO podem ser enviadas diretamente para o MPC ou para os controladores da planta ou passam por um processo de análise dos resultados do otimizador. Neste último caso, as ações de controle devem passar por um teste de significância das movimentações de controle, isto é, verificar através de testes estatísticos se as ações de controle são realmente distintas das anteriores, conforme Figura 2.3.7 (Miletic e Marlin, 1998).

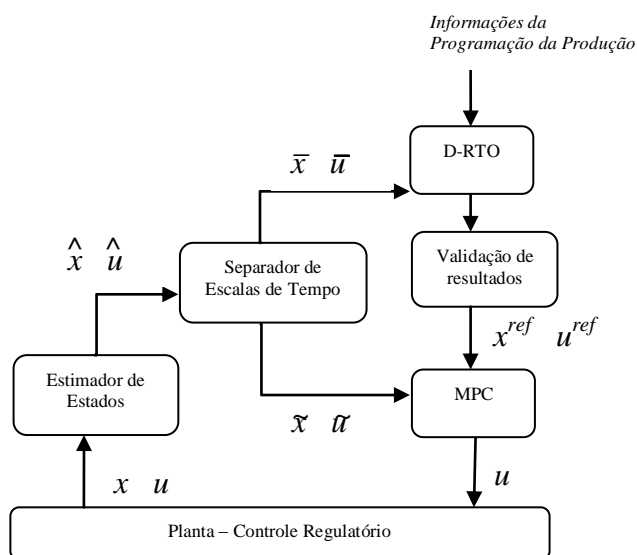


Figura 2.3.7 – Validação de Resultados do D-RTO.

Em suma, há uma série de opções de estrutura de sistema de D-RTO, como consequência das funções necessárias do sistema. A escolha depende da natureza do problema a ser resolvido (complexidade e dinâmica) e da tecnologia disponível para a aplicação.

No presente trabalho, propõe-se estudar e desenvolver um sistema de D-RTO para operar em plantas de processo. Para tanto, será utilizado como planta um módulo de simulação dinâmica de processo, desenvolvido no ambiente EMSO (*Environment for Modeling, Simulation and Optimization*) (Soares e Secchi, 2003), para solução do sistema de equações algébrico-diferenciais, um servidor OPC (*OLE for Process Control*), para armazenar e fornecer os dados de planta, um módulo de otimização dinâmica (D-RTO), um módulo de controle preditivo linear (LMPC), ambos desenvolvidos em Matlab/Simulink, um módulo de separação de escala de tempo, compatíveis com o horizonte de tempo do D-RTO ou com o horizonte de predição do MPC.

Será apresentado ainda um novo algoritmo contendo critérios para ativação do D-RTO (disparador do D-RTO). Este algoritmo é executado em tempo de simulação para atualização das trajetórias de referência das variáveis de otimização no controlador, permitindo assim que o módulo de otimização seja executado somente em situações necessárias. Por limitações de tempo, não serão utilizados neste trabalho os módulos de estimação de estados e de validação de resultados.

## 3 Metodologia do Trabalho

### 3.1 Otimização Dinâmica em Tempo Real

A otimização dinâmica em tempo real (D-RTO) vem contribuir com a solução dos problemas de otimização da indústria química, pela sua capacidade de otimizar o processo, executando transições na operação com menor quantidade de produtos fora de especificação e em menor tempo, reduzindo perdas de produtos, insumos e energia, bem como a produção de efluentes. Além disso, a D-RTO tem a capacidade de controlar e otimizar as receitas de processos em batelada, além de proporcionar estudos de transições.

A D-RTO minimiza um ou mais objetivos de produção, como os descritos acima, manipulando as variáveis de controle (por exemplo: *set-points* de controladores ou até mesmo definindo os pontos ótimos para a camada de controle avançado) ao longo do horizonte de otimização, respeitando todos os limites operacionais e dos equipamentos.

Na última década foram desenvolvidas várias ferramentas comerciais para atender a estas demandas da indústria. As ferramentas de simulação e otimização dinâmica, tais como *ASPEN Dynamics*, *gPROMS*, *DYNOPC* e *INCOOP*, se propõem a resolver estes problemas. Contudo, estas ferramentas não são completas para a implementação de um sistema de D-RTO, limitando-se basicamente à simulação dinâmica e otimização, sendo que o *INCOOP* possui uma arquitetura mais próxima à necessária para isto.

Estas mesmas ferramentas podem ser utilizadas em controladores preditivos não-lineares (NMPC – *Nonlinear Model Predictive Control*), para resolver problemas de controle e otimização de processos com dinâmicas não-lineares. Além disso, podem ser usadas para realizar a identificação de modelos não-lineares do processo.

Para facilitar o entendimento da otimização dinâmica e formular com maior clareza o sistema de D-RTO, é apresentada a solução analítica de um problema simples de otimização dinâmica (Almeida e Secchi, 2006).

Considere que se faça uma encomenda de  $B$  unidades de um produto a serem entregues em um tempo  $T$  e deseja-se planejar a produção com mínimo custo, onde:

$x(t)$  – inventário acumulado;

$x(0) = 0$  – quantidade inicial;

$x(T) = B$  – quantidade a ser entregue;

$\dot{x}(t)$  – taxa de produção.

O custo de produção é dado pela composição do custo de fabricação do produto mais o custo de armazenamento, dado por:

$$\varphi(\dot{x}, x, t) = c_1 \dot{x}(t)^2 + c_2 x(t) \quad (3.1.1)$$

Considerando que o modelo do processo seja simplesmente  $\dot{x}(t) = u(t)$ , onde  $u(t)$  é o *setpoint* do controle de vazão de produto, sendo esta a variável de controle do



problema de otimização dinâmica. Portanto, pode-se formular o problema de otimização dinâmica da seguinte forma:

$$\begin{aligned} & \min_u \int_0^T [c_1 \dot{x}(t)^2 + c_2 x(t)] dt \\ & \text{s.a :} \\ & \dot{x}(t) = u(t) \\ & x(0) = 0 \\ & x(T) = B \\ & u(t) \geq 0 \end{aligned} \tag{3.1.2}$$

Para resolver o problema usando a abordagem do controle ótimo (Princípio do Máximo de Pontryagin), precisa-se calcular o Hamiltoniano e estabelecer a condição necessária para otimalidade como a seguir:

$$\begin{aligned} H &= c_1 u(t)^2 + c_2 x(t) + \lambda u(t) \\ \frac{\partial H}{\partial u} &= 2c_1 u(t) + \lambda = 0 \therefore u(t) = \frac{-\lambda}{2c_1} \\ -\frac{\partial H}{\partial x} &= \dot{\lambda} = -c_2 \end{aligned} \tag{3.1.3}$$

Deve-se resolver

$$\begin{aligned} \dot{x}(t) &= u(t) & x(0) &= 0 \\ \dot{\lambda} &= -c_2 & x(T) &= B \\ u(t) &= \frac{-\lambda}{2c_1} \end{aligned} \tag{3.1.4}$$

Primeiramente integra-se  $\dot{\lambda}$ , portanto:

$$\lambda(t) = -c_2 t + k_1 \quad (3.1.5)$$

Substituindo  $\lambda(t)$  na condição de otimalidade

$$u(t) = \frac{-\lambda}{2c_1} = \frac{c_2 t - k_1}{2c_1} \quad (3.1.6)$$

Substituindo  $u(t)$  na equação de estado e integrando, obtém-se:

$$\dot{x}(t) = \frac{c_2 t - k_1}{2c_1} \quad (3.1.7)$$

$$x(t) = \frac{c_2 t^2}{4c_1} - \frac{k_1 t}{2c_1} + k_2 \quad (3.1.8)$$

Obedecendo às condições de contorno, obtém-se:

$$\begin{aligned} x(0) = k_2 = 0 \quad \therefore \quad k_2 = 0 \\ x(T) = \frac{c_2 T^2}{4c_1} + \frac{-k_1 T}{2c_1} = B \quad \therefore \quad k_1 = \frac{c_2 T}{2} - \frac{2c_1 B}{T} \end{aligned} \quad (3.1.9)$$

Substituindo nas equações, obtém-se:

$$\begin{aligned} x(t) &= \frac{c_2 t}{4c_1} (t - T) + \frac{B}{T} t \\ u(t) &= \frac{c_2}{4c_1} (2t - T) + \frac{B}{T} \\ \lambda(t) &= -c_2 t + \frac{c_2 T}{2} - \frac{2c_1 B}{T} \end{aligned} \quad (3.1.10)$$

que correspondem às formas funcionais que minimizam a função objetivo ao longo do horizonte de otimização  $T$ .

### 3.1.1 Formulação do Problema de Otimização Dinâmica

Para resolver problemas de otimização dinâmica, necessita-se primeiramente formular o problema (Figura 3.1.1.1). O problema de otimização varia conforme o caso

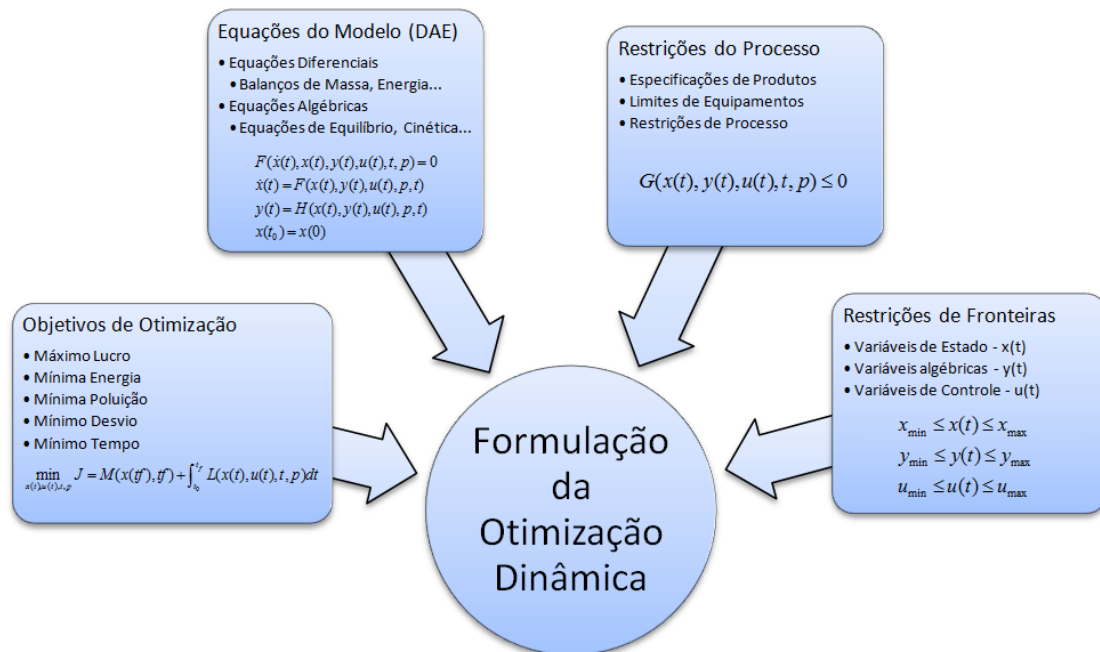


Figura 3.1.1.1 – Formulação do Problema de Otimização Dinâmica.

e o sistema a ser otimizado, e é constituído basicamente de uma função objetivo ou até mesmo multi-objetivos, restrição de igualdade (modelo do processo e restrições adicionais), restrições de desigualdade (que define a região viável) e restrições de fronteiras (que definem os domínios das variáveis de estado, de saída e de controle).

### 3.1.1.1 O Problema Geral

O problema de otimização dinâmica contínuo tem a seguinte formulação geral:

$$\min_{x(t), u(t), t_f, p} J(x(t), u(t), t, p) = \phi(x(t_f), t_f) + \int_{t_0}^{t_f} \varphi(x(t), u(t), t, p) dt$$

Sujeito a:

$$\begin{aligned} F(\dot{x}(t), x(t), y(t), u(t), t, p) &= 0 && \text{Modelo Dinâmico (ODE)} \\ H(x(t), y(t), u(t), t, p) &= 0 && \text{Equações Algébricas (AE)} \\ G(x(t), y(t), u(t), t, p) &\leq 0, \quad \forall t \in T && \text{Restrições de Desigualdade} \\ I(x_0, y_0, u_0, t_0, p) &= 0 && \text{Condições Iniciais} \\ x_{\min} &\leq x(t) \leq x_{\max} \\ y_{\min} &\leq y(t) \leq y_{\max} \\ u_{\min} &\leq u(t) \leq u_{\max} && \text{Fronteiras} \\ p_{\min} &\leq p \leq p_{\max} \\ t_f^{\min} &\leq t_f \leq t_f^{\max} \end{aligned}$$

(3.1.1.1.1)

Onde:

$T$  – domínio do tempo  $[t_0, t_f]$ ;

$J$  – função objetivo escalar;

$F$  – equações diferenciais;

$H$  – equações algébricas;

$G$  – inequações algébricas;

$x$  – vetor de variáveis de estado ( $x \in \mathfrak{R}^{nx}$ );

$x_0$  – vetor de condições iniciais das variáveis de estado;

$y$  – vetor de variáveis algébricas ( $y \in \mathfrak{R}^{ny}$ );

$u$  – vetor de variáveis de controle ( $u \in \mathfrak{R}^{mu}$ );

$p$  – vetor de parâmetros independente do tempo.

Ao formular o problema de otimização dinâmica, deve-se analisar o sistema quanto ao seu objetivo ( $J$ ), quanto ao modelo do processo ( $F$ ) e suas restrições e fronteiras ( $x$ ,  $y$ ,  $u$  e  $p$ ). Na definição dos objetivos, devem ser definidos os benefícios a serem obtidos do sistema e a métrica de avaliação do desempenho. Além disso, deve-se definir quais são as variáveis de otimização e de decisão envolvidas no problema. Na formulação do problema de otimização, deve-se procurar definir as restrições de processo, segurança, e outras ( $G$ ). Além disso, deve-se definir a região de operação do sistema ( $H$ ), que influencia diretamente na região viável de solução do problema de otimização. Neste processo, deve-se inicialmente definir o fluxograma de processo do sistema, coletar dados do processo e estabelecer as hipóteses ligadas ao problema de otimização.

Uma vez definida a estrutura básica do problema de otimização, deve-se construir o modelo do processo, definindo quais tipos de equações diferenciais devem ser usadas (ex.: balanços de massa, energia *etc.*), quais equações algébricas devem estar no modelo (ex.: equações de equilíbrio, cinéticas *etc.*). Com isso, têm-se a definição de quantas variáveis de otimização e equações estão envolvidas no problema. Desta forma,

tem-se uma avaliação dos graus de liberdade do sistema para definir as variáveis de decisão do mesmo.

As variáveis do sistema têm restrições de fronteira, tanto nas variáveis de controle, que estão relacionadas a atuadores (ex.: vazões, pressões *etc.*), como nas variáveis de estado, que estão ligadas a restrições operacionais de segurança ou desempenho (ex.: temperatura, recuperação), bem como limites de especificação de produtos (ex.: pureza, concentração).

Além disso, podem-se ter restrições terminais, ligadas a desempenho, tais como: conversão, volume total produzido, dentre outras.

Uma vez que a estrutura do problema e o modelo do processo estão definidos, deve-se determinar qual será o método de solução do problema, bem como o algoritmo de otimização a ser utilizado. Ao definir estes pontos, é possível que haja a necessidade de reformular a apresentação do problema de otimização (ex.: discretizar o problema) para que possa utilizar o algoritmo escolhido.

A seguir, é apresentado como devem ser definidas as partes e as formas do problema de otimização dinâmica.

### **3.1.1.2 *Formulação da Função Objetivo***

A função objetivo é uma medida escalar do desempenho ou sucesso de um determinado processo a ser otimizado (maximizado ou minimizado). Os objetivos típicos de um processo a ser otimizado são normalmente ligados a critérios econômicos

(ex.: maximizar o lucro ou minimizar os custos; minimizar o consumo de insumos e energia), qualidade de produtos (ex.: minimizar folga de especificação de produtos), flexibilidade (ex.: minimizar o tempo de uma transição de processo ou de uma batelada), e critérios ambientais (ex.: minimizar a produção de efluentes poluentes).

Há casos onde se deseja atender a várias especificações e requisitos, tais como máximo lucro, mínima folga de especificação de produtos e máxima vazão de carga. Há situações onde os objetivos são concorrentes, e deve-se obter uma solução de compromisso. Neste caso, a formulação de uma única função objetivo com restrições pode não representar de forma adequada o problema, sendo necessária a definição de um conjunto de funções objetivo, que devem ser balanceadas de alguma maneira.

Genericamente, um índice de desempenho pode ter três formas possíveis. A formulação mais genérica é apresentada na forma do *Problema de Bolza*, onde se tem uma parcela referente ao estado final do processo, chamado termo de *Mayer*, e outra ligada ao desempenho ao longo da trajetória do sistema, chamado termo de *Lagrange* (Bliss, 1946; Kamien e Schwartz, 1991). Desta forma, pode-se apresentar o problema de *Bolza* na forma abaixo:

### **Problema de Bolza**

$$J(u(t)) = \underbrace{\phi(x(t_f), t_f)}_{\text{termo de Mayer}} + \underbrace{\int_{t_0}^{t_f} \varphi(x(t), u(t), t) dt}_{\text{termo de Lagrange}} \quad (3.1.1.2.1)$$

Há dois casos especiais que são conseqüências de simplificações do *Problema de Bolza*, são elas as formas de *Lagrange* e de *Mayer*.

### **Problema de Lagrange**

Na forma do *Problema de Lagrange* só interessa a parte integral da função objetivo. Desta forma, pode-se dizer que o *Problema de Lagrange* é o *Problema de Bolza* onde  $\phi(x(t_f), t_f) \equiv 0$ . Portanto, a função objetivo passa a ter a seguinte forma:

$$J(u(t)) = \int_{t_0}^{t_f} \varphi(x(t), u(t), t) dt \quad (3.1.1.2.2)$$

onde:

$J$  – critério de desempenho da otimização;

$\phi$  – parte que avalia a condição final da função objetivo;

$\varphi$  – parte que avalia a função objetivo ao longo do horizonte de otimização;

$x$  – vetor das variáveis de estado;

$u$  – vetor das variáveis de controle.

### **Problema de Mayer**

Por outro lado, na forma do *Problema de Mayer* só interessa a parte da função objetivo relativo ao estado final do sistema. Desta forma, pode-se dizer que o *Problema de Mayer* é o *Problema de Bolza* onde  $\varphi(x(t), u(t), t) \equiv 0$  e, portanto, a função objetivo passa a ter a seguinte forma:



$$J(u(t)) = \phi(x(t_f), t_f) \quad (3.1.1.2.3)$$

Na realidade, os problemas de *Mayer* e de *Lagrange* são tão gerais quanto o *Problema de Bolza*. O *Problema de Bolza* pode ser formulado usando estados adicionais, podendo-se definir um novo vetor de estados  $z$  tal que,

$$z(t) = [x^0(t) \quad x(t)^T]^T = [x^0(t) \quad x^1(t) \quad \cdots \quad x^{nx}(t)]^T \quad (3.1.1.2.4)$$

onde  $z(t) \in \mathfrak{R}^{nx+1}$ ,  $t \in [t_0, t_f]$  e  $x_0(\cdot)$  é uma função contínua tal que:

$$\dot{x}^0(t) = \varphi(x(t), u(t), t), \quad x^0(t_0) = 0 \quad (3.1.1.2.5)$$

Desta forma

$$x^0(t) = \int_{t_0}^t \varphi(x(\tau), u(\tau), \tau) d\tau \quad (3.1.1.2.6)$$

e o *Problema de Bolza* pode ser escrito como

$$J(u(t)) = \phi(x(t_f), t_f) + x^0(t_f) \quad (3.1.1.2.7)$$

sendo que se deve acrescentar uma restrição de igualdade e uma condição inicial no problema de otimização, conforme equação (3.1.1.2.5).

Da mesma forma, um *Problema de Bolza* pode ser transformado em um *Problema de Mayer*.

Também se pode transformar um *Problema de Bolza* em *Problema de Lagrange*. Se formular o *Problema de Bolza* usando estados adicionais, então se pode definir um novo vetor de estados  $z$  onde:

$$z(t) = [x_0(t) \quad x(t)^T]^T = [x_0(t) \quad x_1(t) \quad \dots \quad x_N(t)]^T \quad (3.1.1.2.8)$$

sendo  $z(t) \in \mathfrak{R}^{N+1}$ ,  $t \in [t_0, t_f]$  e  $x_0(\cdot)$  é uma função contínua tal que:

$$x_0(t) = \frac{d\varphi(x(t), u(t), t)}{dt} \quad \text{e} \quad \dot{x}_0(t) = 0 \quad (3.1.1.2.9)$$

Desta forma

$$\varphi(x(t_f), u(t_f), t_f) = \int_{t_0}^{t_f} x_0(t) dt \quad (3.1.1.2.10)$$

pode-se reescrever o *Problema de Bolza*

$$J(u(t)) = \int_{t_0}^{t_f} \{ \varphi(x(t), u(t), t) + x_0(t) \} dt \quad (3.1.1.2.11)$$

### **Tipos de Função Objetivo**

Na otimização dinâmica de processos, há uma grande variedade de tipos de função objetivo, como por exemplo as empregadas em controle ótimo, MPC, detecção de erros grosseiros, reconciliação de dados e ajuste de parâmetros, e D-RTO. Uma descrição mais detalhada pode ser encontrada em Almeida e Secchi (2006).

Em aplicações de D-RTO, os problemas possuem uma grande diversidade de objetivos, em função da sua aplicação, ou seja, o tipo de sistema (ex.: reator em batelada, torre de destilação contínua *etc.*) e sua finalidade (ex.: minimizar consumo de energia, minimizar o tempo de transição da operação). Há uma série de objetivos comuns em processos químicos:

- Alcançar o estado do sistema o mais próximo possível de um valor desejado no tempo final  $t_f$  ;
- Minimizar o tempo para atingir um determinado estado desejado e mantê-lo na posição (minimiza o tempo de transição).
- Minimizar a energia para atingir um determinado estado desejado em  $t_f$  ;
- Maximizar ou minimizar índices de desempenho específicos como lucro, recuperação de produto, eficiência, produção total.

Em função desta grande diversidade de objetivos, não há formas típicas de função objetivo em problemas de D-RTO, a não ser que tenha apenas finalidade de controle ótimo. Portanto, deve-se utilizar a função objetivo na sua forma geral de *Bolza*.

$$J(x(t), u(t), t) = \phi(x(t_f), u(t_f), t_f) + \int_{t_0}^{t_f} \varphi(x(t), u(t), t) dt \quad (3.1.1.2.12)$$

É comum encontrar a função de *Mayer* aplicada em problemas de otimização dinâmica, sendo que o termo da função de *Lagrange* é normalmente convertido no termo de *Mayer*. Portanto a forma mais geral de formular a função objetivo na forma de problema de *Mayer* é a seguinte:

$$J(x(t), u(t), t) = \sum_{i=1}^{NO} \varphi_i(x(t_f), u(t_f), t_f) \quad (3.1.1.2.13)$$

onde  $\varphi_i(x, u, t_f)$  é o termo de Mayer do índice de desempenho  $i$  e  $NO$  é o número de objetivos.

### **3.1.1.3 Formulação das Restrições**

Em um problema de otimização, têm-se os seguintes tipos de restrições:

- Restrições de Igualdade – Modelo dinâmico do processo;
- Restrições de Desigualdade – Limitações do Processo.

#### **Restrições de Igualdade – Formulação do Modelo do Processo**

Os problemas de otimização dinâmica têm dois tipos de restrições de igualdade, o próprio modelo dinâmico do processo, que define as trajetórias das variáveis de estado e restrições impostas em determinados instantes de tempo ao longo do horizonte de otimização ou no instante final.

Como as trajetórias das variáveis de estado são definidas pelo modelo dinâmico do processo, é muito importante que se mantenha o modelo o mais fiel possível à planta real. O modelo deve considerar os aspectos macroscópicos ou microscópicos do processo, pode conter equações de propriedades físicas e outras restrições de igualdade. O modelo também deve descrever os aspectos de segurança, qualidade de produto, eficiência de equipamento, controladores, dentre outros. Além disso, deve-se ter em

mente que o modelo irá exigir um esforço computacional por parte do otimizador e também deverão ser evitados problemas de convergência. Portanto, deve-se utilizar o modelo na forma mais simples possível, com qualidade necessária, para resolver o problema de otimização e que tenha o mínimo de problemas de estabilidade numérica. Em alguns casos, isto exige que se faça uma redução do modelo.

Processos químicos são geralmente modelados dinamicamente usando equações diferenciais originadas de leis de conservação (balanços de massa, de energia e de quantidade de movimento) e por equações algébricas que definem relações físicas e termodinâmicas, além de equações constitutivas do processo, formando um conjunto de equações algébrico-diferenciais (DAEs – *Differential-Algebraic Equations*). O modelo pode ser representado na forma de equações explícitas, semi-implícitas e implícitas.

A forma na qual o modelo deve ser apresentado está diretamente ligado à maneira com a qual o algoritmo de otimização foi implementado.

O modelo pode ser representado na forma de equações explícitas:

$$\begin{aligned} \dot{x}(t) &= F(x(t), y(t), u(t), t, p) \\ y(t) &= H(x(t), y(t), u(t), t, p) \\ x(t_0) &= x(0) \\ y(t_0) &= y(0) \end{aligned} \tag{3.1.1.3.1}$$

Pode ser representado na forma de equações semi-implícitas:

$$\begin{aligned}
\dot{x}(t) &= F(x(t), y(t), u(t), t, p) \\
H(x(t), y(t), u(t), t, p) &= 0 \\
x(t_0) &= x(0) \\
y(t_0) &= y(0)
\end{aligned}
\tag{3.1.1.3.2}$$

Ou representado na forma de equações implícitas:

$$\begin{aligned}
F(\dot{x}(t), x(t), y(t), u(t), t, p) &= 0 \\
H(x(t), y(t), u(t), t, p) &= 0 \\
x(t_0) &= x(0) \\
y(t_0) &= y(0)
\end{aligned}
\tag{3.1.1.3.3}$$

Caso o algoritmo de otimização utilize uma forma diferente do modelo disponível, deve-se transformá-lo na forma adequada a esse algoritmo. Por exemplo, se o modelo estiver representado na forma explícita, transforma-se na forma implícita apenas colocando a equação diferencial na sua forma residual.

Seja o sistema de equações diferenciais do modelo na forma explícita

$$\dot{x}(t) = F(x(t), y(t), u(t), t, p)
\tag{3.1.1.3.4}$$

Transforma-se na forma implícita, colocando todos os termos no lado direito da equação diferencial.

$$0 = F(x(t), y(t), u(t), t, p) - \dot{x}(t)
\tag{3.1.1.3.5}$$

Se o modelo estiver na forma implícita e o algoritmo de otimização utiliza o modelo na forma explícita ou semi-implícita, deve-se adicionar uma variável de estado algébrica. Por exemplo, considere o sistema de primeira ordem implícito

$$F(\dot{x}(t), x(t), y(t), u(t), t, p) = 0 \quad (3.1.1.3.6)$$

Este sistema implícito é equivalente ao sistema de equações diferenciais explícito e equações algébricas não-lineares na forma

$$\begin{aligned} \dot{x}(t) &= v(t) \\ 0 &= F(x(t), y(t), u(t), t, p) \end{aligned} \quad (3.1.1.3.7)$$

onde o vetor de variáveis de estado algébricas aumentado é  $y = [y \ v]^T$ .

Normalmente, os modelos dinâmicos resultam em sistemas DAE de índice 1, isto é, as variáveis algébricas podem ser resolvidas diretamente pelas equações algébricas. Caso o problema tenha índice superior, recomenda-se que o sistema DAE seja reformulado para o índice 1 (Petzold, 1989; Costa *et al.*, 2003), ou então utilizar algoritmos de integração próprios para sistemas de índice superior.

O sistema DAE pode conter equações diferenciais de ordem superior. Neste caso, as equações de ordem superior devem ser transformadas em equações diferenciais de primeira ordem, pois a maioria dos algoritmos de otimização dinâmica resolve apenas sistema de equações diferenciais de primeira ordem. Um sistema de equações diferenciais de ordem superior pode ser transformado em primeira ordem introduzindo um vetor de variáveis de estado aumentado.

Considere um sistema de segunda ordem na forma semi-implícita

$$\ddot{x}(t) = f(x(t), u(t), p, t) \quad (3.1.1.3.8)$$

Este sistema é equivalente ao sistema de primeira ordem na forma

$$\begin{aligned} \dot{x}(t) &= v(t) \\ \dot{v}(t) &= f(x(t), u(t), p, t) \end{aligned} \quad (3.1.1.3.9)$$

onde o vetor de variáveis de estado aumentado é  $x = [x \ v]^T$ .

### **Restrições de Desigualdade – Limitações do Processo**

Muitas vezes o problema de otimização tem sua região viável de operação limitada em função de restrições operacionais ou segurança. Estas restrições podem ser impostas às variáveis de estado ou de controle envolvidas no problema. Também podem estar distribuídas no tempo, podendo ser restrições ao longo do horizonte de otimização ou no instante final (restrições terminais).

As restrições podem ser referentes a variáveis ligadas diretamente ao objetivo de otimização (ex.: máximo investimento permitido), outras variáveis ligadas a limitações de equipamentos (ex.: máxima vazão da bomba), a aspectos de segurança (ex.: máxima pressão no reator), ao meio ambiente (ex.: máxima carga de poluentes permitida). Além disso, têm-se restrições de domínios das variáveis de estado e controle bem como dos parâmetros do modelo.



Normalmente, são representados na forma de  $g(x) \leq 0$ , e podem ser convertidas em  $g(x) \geq 0$  multiplicando a restrição por -1.

Dentre as mais variadas formas de restrições de desigualdade em um problema de otimização dinâmica, destacam-se as seguintes:

### **Restrições de domínios das variáveis**

As restrições de domínio podem se referir as variáveis de controle (de decisão), variáveis de estado diferenciais ou algébricas.

#### Limites nas variáveis de controle

Normalmente as variáveis de controle são as variáveis manipuladas dos sistemas de controle digitais, e os seus limites são expressos como:

$$u_{\min} \leq u(t) \leq u_{\max}, \quad \forall t \in T \quad \text{onde } T = [t_0, t_f] \quad (3.1.1.3.10)$$

#### Limites no tempo

Nos problemas de otimização onde o tempo final está livre, a variável tempo passa a ser uma variável de estado e seu domínio viável é:

$$t_{\min} \leq t \leq t_{\max}, \quad \forall t \in T \quad \text{onde } T = [t_0, t_f] \quad (3.1.1.3.11)$$

#### Limites nas variáveis de estado (diferenciais) e de saída (algébricas)

O domínio das variáveis dependentes é limitado em:

$$\begin{aligned} x_{\min} \leq x(t) \leq x_{\max} \\ y_{\min} \leq y(t) \leq y_{\max} \end{aligned}, \quad \forall t \in T \quad \text{onde } T=[t_0, t_f] \quad (3.1.1.3.12)$$

### **Restrições terminais das variáveis**

É comum estabelecer restrições deste tipo no problema, pois normalmente se especificam os valores mínimos e máximos para as variáveis de controle e de estado. Pode-se ter restrições terminais de igualdade e desigualdade.

$$\begin{aligned} h(x(t_f), u(t_f), t_f) &= 0 \\ g(x(t_f), u(t_f), t_f) &\leq 0 \end{aligned} \quad (3.1.1.3.13)$$

Há problemas de otimização onde se deseja que algumas variáveis de estado, de controle ou uma composição delas tenham uma determinada posição no tempo final ou que as mesmas estejam em determinados intervalos no instante final.

$$\begin{aligned} v(t_f) &= v^* \quad \text{ou} \\ v_{\min} &\leq v(t_f) \leq v_{\max} \end{aligned} \quad (3.1.1.3.14)$$

### **Restrições Interiores das variáveis**

Há situações em que se deseja limitar algumas variáveis em alguns instantes de tempo específicos dentro do horizonte de otimização. Algumas receitas de processo podem exigir que se tenha um determinado estado ou que estejam num determinado

intervalo em certos momentos do processo. Estas restrições passam a ser apresentadas como:

$$\begin{aligned}
 v(t_I) &= v^* \quad \text{ou} \\
 v_{\min}(t_I) &\leq v(t_I) \leq v_{\max}(t_I) \\
 T &= \{t_1, t_2, \dots, t_M\} \quad \text{ou} \quad T = \{[\tau_0, \tau_f]_1, \dots, [\tau_0, \tau_f]_{NI}\}
 \end{aligned}
 \tag{3.1.1.3.15}$$

onde  $t_I$  é um instante de tempo dentro do horizonte de otimização.

### Restrições nas trajetórias das variáveis

Há casos em que se deseja limitar determinadas variáveis ao longo de todo horizonte de otimização, ou seja:

$$g(x(t), u(t), t) \leq 0, \quad t \in [t_0, t_f]$$

Ou

$$v_{\min} \leq v(t) \leq v_{\max}, \quad t \in [t_0, t_f]$$

### 3.1.2 Solução do Problema de Otimização Dinâmica

Problemas de otimização dinâmica (DAOP – *Differential Algebraic Optimization Problems*), de dimensão infinita, podem ser resolvidos por duas classes de métodos: diretos e indiretos. Esta classificação foi inicialmente atribuída a Edelbaum (1962).

Os métodos indiretos resolvem o problema de otimização dinâmica através da satisfação das condições necessárias de primeira ordem para otimalidade. Estas

condições são originárias da teoria de controle ótimo e aplicação do cálculo variacional. Estes métodos também são conhecidos como métodos analíticos pelo fato de utilizar as condições de otimalidade para transformar o DAOP em um problema de valor de contorno duplo (TPBVP – *Two Point Boundary Value Problem*), que era resolvido analiticamente, por busca direta ou até mesmo graficamente. Há, basicamente, dois métodos importantes usando este tipo de abordagem:

- **Programação Dinâmica** (Bellman, 1957) – Baseado no princípio da otimalidade de Bellman e na solução das Equações de Hamilton-Jacobi-Bellman (HJB);
- **Princípio do Máximo de Pontryagin** (Pontryagin, 1962) – Baseado na solução das condições de otimalidade da teoria de controle ótimo.

Os métodos numéricos que resolvem o TPBVP, isto é, integram o sistema de equações diferenciais com valor de contorno duplo, são comumente encontrados na literatura como: métodos tipo *shooting*<sup>1</sup> (*single shooting* ou *multiple shooting*), métodos de quadratura (diferenças finitas, métodos dos resíduos ponderados, colocação em elementos finitos), método *invariant embedding* ou programação dinâmica. Uma revisão dessas abordagens pode ser encontrada em Cervantes e Biegler (1998).

Os métodos indiretos de solução de problemas de otimização dinâmica não são adequados para resolver problemas complicados de dimensão elevada. Por esta razão, estes métodos têm sido mais usados nas áreas aeronáutica e espacial do que na indústria de processos.

---

<sup>1</sup> O método *single shooting* surgiu da necessidade de se resolver os problemas de controle ótimo na área de balística, onde se desejava calcular a trajetória ótima de mísseis do ponto de lançamento para atingir um determinado alvo. Daí vem o jargão utilizado “*shooting*”.

Como alternativa aos métodos indiretos, tem-se os métodos diretos ou numéricos. Estes métodos não utilizam as condições de otimalidade para resolver o DAOP, mas sim aplicam algum nível de discretização diretamente no DAOP, transformando-o em um problema de programação não-linear (NLP – *Non-Linear Programming*) de dimensão finita. O problema de NLP, então, pode ser resolvido por alguns dos algoritmos disponíveis na literatura (ex.: SQP, rSQP, dentre outros). A escolha do algoritmo apropriado dependerá da dimensão e da natureza do problema de NLP a ser resolvido.

Há, basicamente, dois níveis de discretização de variáveis, que subdividem os métodos diretos em dois grupos: métodos de discretização parcial e total (vide Figura 3.1.2.1).

Nos métodos de discretização parcial, apenas as variáveis de controle são discretizadas, e as variáveis de estado são obtidas por integração do sistema de equações algébrico-diferenciais. Neste caso, define-se uma forma funcional para as variáveis de controle, escolhem-se os pontos discretos no intervalo de integração e um perfil inicial para estas variáveis, integra-se o modelo neste intervalo e com as variáveis de estado obtidas e as sensibilidades da função objetivo, o algoritmo de NLP procura a solução ótima.

Nesta abordagem, somente as variáveis de controle são tratadas como variáveis de otimização pelo algoritmo de NLP, pois as variáveis de estado são resolvidas por integração do sistema DAE. Por esta razão, estes métodos também são chamados de métodos *sequenciais*, pois se resolvem o problema de otimização em uma seqüência de integração e otimização. Dentre os métodos de discretização parcial encontrados na

literatura, pode-se destacar a programação dinâmica e os métodos de *shooting* (*single shooting* e *multiple-shooting*). Nos métodos de *multiple-shooting*, também é realizada uma partição do intervalo de integração em subintervalos menores para poder explorar propriedades de processamento paralelo e reduzir problemas numéricos em sistemas instáveis.

Nos métodos de discretização total, além de discretizar as variáveis de controle, discretizam-se também as variáveis de estado, resultando em um sistema de equações algébricas não-lineares (NLA – *Non-Linear Algebraic equations*), em que são resolvidos, simultaneamente, a integração do sistema DAE e o problema de NLP. Por esta razão, estes métodos também são chamados de métodos *simultâneos*. Dentre os métodos de discretização total, encontrados na literatura, podem-se destacar os métodos de colocação (métodos dos resíduos ponderados), diferenças finitas e colocação em elementos finitos.

Os problemas de NLP gerados pelos métodos diretos simultâneos são de dimensão elevada. Felizmente, uma série de algoritmos de NLP (ex.: SQP, rSQP) tem a capacidade de resolver problemas de otimização dinâmica dessas dimensões. Por isso, estes métodos têm sido objeto de interesse da indústria de processos, resolvendo problemas de otimização dinâmica para DRTO e NMPC.

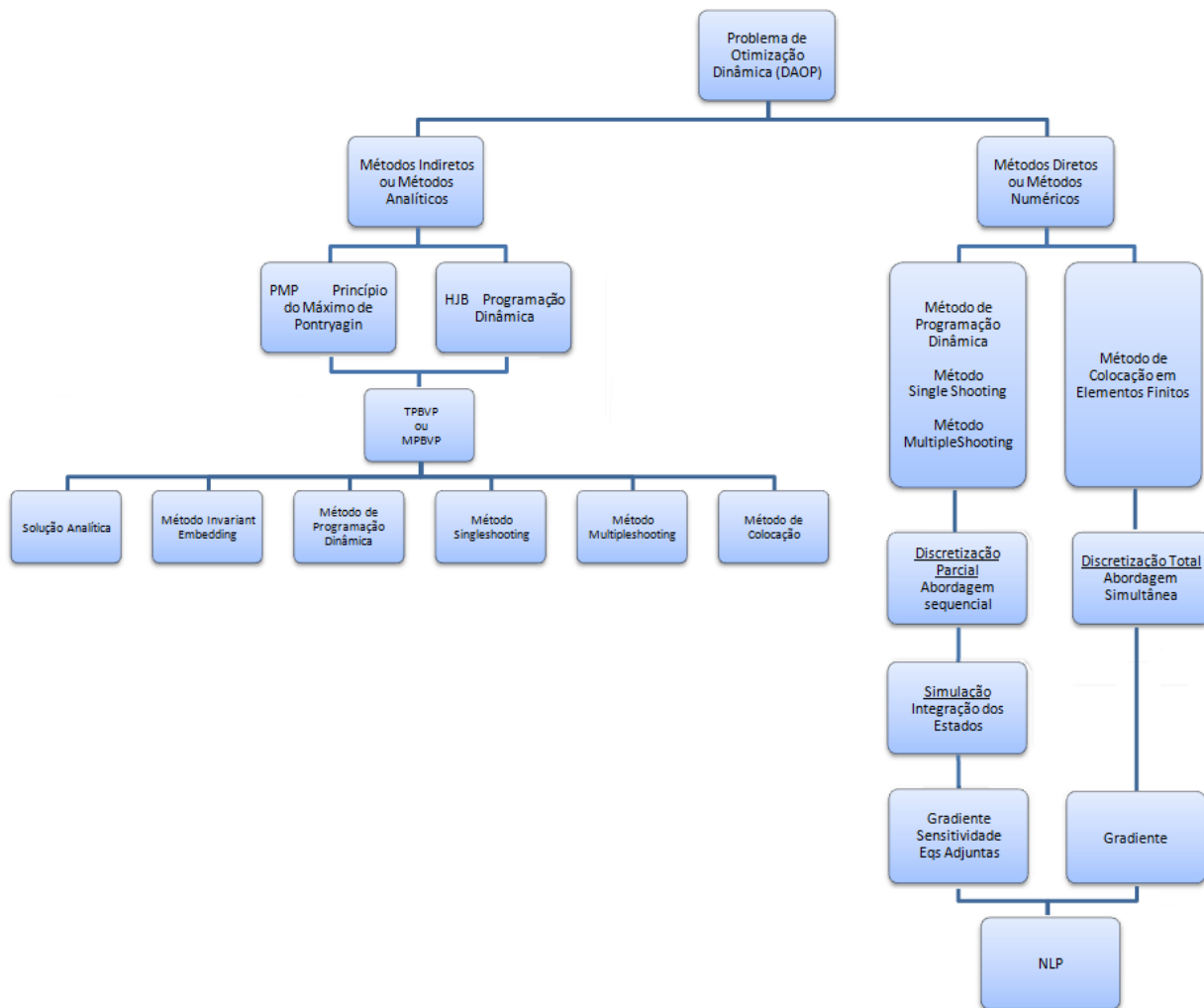


Figura 3.1.2.1 – Métodos de Solução de problemas de otimização dinâmica (DAOP).

### 3.2 Estrutura Proposta do Sistema de Otimização Dinâmica em Tempo Real

A otimização dinâmica em tempo real (D-RTO), conforme visto no Capítulo 2, é uma aplicação *on-line*, sendo executada de forma cíclica e conectada à planta, onde se coletam informações de instrumentos da planta e se realiza o cálculo da trajetória ótima das variáveis de controle, com o auxílio de algoritmos de otimização dinâmica. Os valores das ações de controle calculadas, por sua vez, são implementados na planta, alterando o ponto de operação.

Com o objetivo de contribuir com a consolidação desta tecnologia, neste trabalho se propõe desenvolver e estudar um sistema de D-RTO para operar em plantas de processo. A estrutura será formada pelos seguintes módulos:

- Módulo de simulação dinâmica de processo (a planta), para solução do sistema de equações algébrico-diferenciais;
- Servidor OPC (*OLE for Process Control*), para armazenar e fornecer os dados de planta;
- Módulo de otimização dinâmica (D-RTO);
- Módulo de controle preditivo linear (LMPC);
- Módulo do Disparador D-RTO;

A Figura 3.2.1 mostra um esquema simplificado da estrutura de D-RTO.

Neste trabalho, por limitação de tempo, o módulo de estimação de estados (EKF – *Extended Kalman Filter*) não será utilizado (as variáveis de estado são consideradas iguais às variáveis medidas e não há ruído, pois a planta é simulada).

A vantagem da estrutura D-RTO/MPC em camadas distintas é que se fornece uma separação clara entre as tarefas das camadas de controle e otimização. Esta separação é realizada com respeito às escalas de tempo assim como aos modelos. Modelos não-lineares rigorosos são usados somente na camada de otimização dinâmica. Tais modelos hoje em dia estão geralmente disponíveis a partir da fase de projeto da planta, logo o esforço adicional para desenvolver o modelo não é tão alto (Engell,



2006). Os algoritmos de controle são geralmente baseados em modelos lineares os quais podem ser determinados a partir de dados da planta ou por linearização de modelos não-lineares.

A otimização dinâmica é usada para projetar estratégias operacionais de processos em batelada, semi-batelada e contínuos durante as fases transientes, tais como: transições de processo, partida ou parada de sistemas e otimização, controle e monitoração de processos em tempo real.

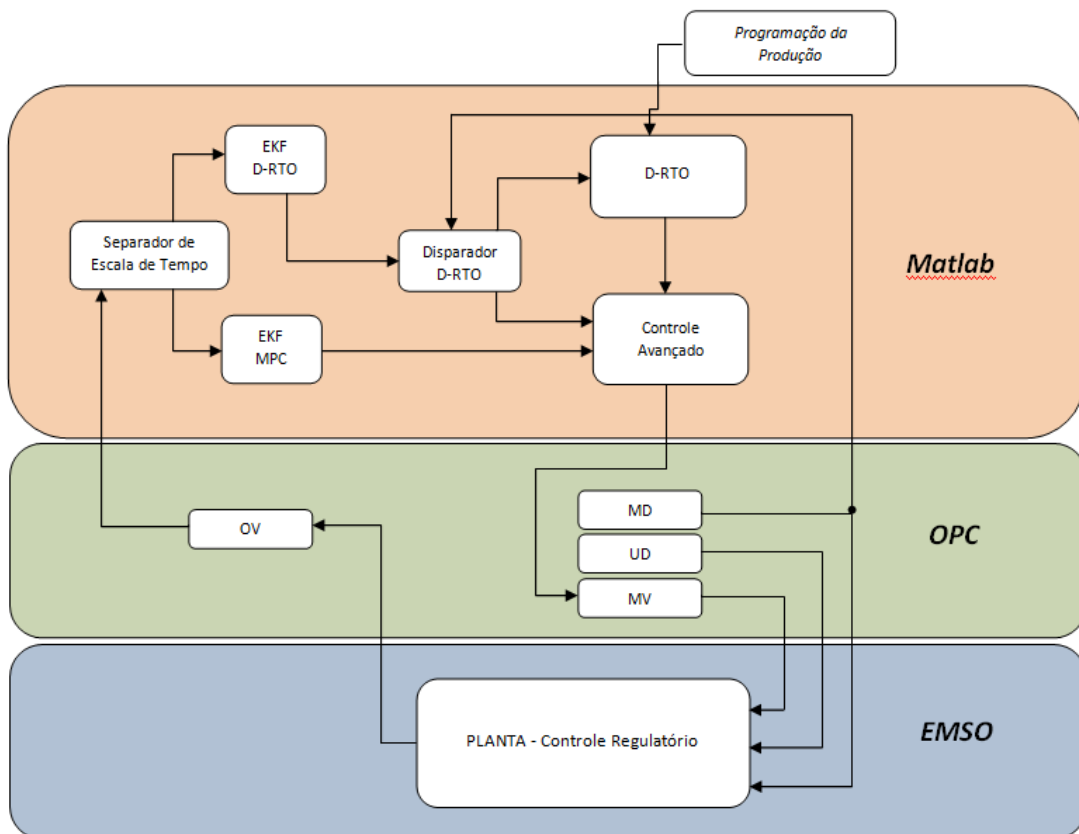


Figura 3.2.1 – Estrutura simplificada do sistema de D-RTO. OV – Variáveis de saída da planta; MD – Perturbações medidas; UD – Perturbações não medidas; MV – Variáveis manipuladas.

### **3.2.1 Módulo do Servidor OPC**

O módulo do servidor OPC serve para armazenar e fornecer os dados da planta. Nele são configurados os *tags* referentes a todas as variáveis do processo. A tecnologia OPC foi desenvolvida para conectar ferramentas computacionais com equipamentos de controle de processos. O OPC é um protocolo de comunicação aberto que permite um método consistente de acesso aos dados de inúmeros equipamentos dos mais diversos fabricantes. O método é o mesmo, independente da origem dos dados, o que vem oferecer ao usuário final uma maior liberdade na escolha dos equipamentos independentemente da disponibilidade de *drivers* de comunicação proprietários. O OPC é construído usando tecnologia Microsoft OLE/COM, mas a especificação OPC foi desenvolvida por uma fundação aberta, a *OPC Foundation*, para atender às necessidades gerais da indústria e não às necessidades específicas de alguns fabricantes de *hardware* e *software* (*OPC Foundation*, 1988). A especificação ainda prevê a evolução das funcionalidades ao longo do tempo e, por isso, os componentes OPC podem se manter no topo das necessidades emergentes da indústria.

Os componentes OPC se classificam em duas categorias: Clientes OPC e Servidores OPC. Um Cliente OPC é tipicamente um usuário dos dados tais como uma Interface de Operação ou um Sistema Supervisório (SCADA). Um Servidor OPC é uma fonte de dados que coleta ou gera dados a partir de um processo, disponibilizando-os aos Clientes OPC. O Cliente OPC interage com o Servidor OPC usando uma interface bem definida. Qualquer Cliente OPC pode se comunicar com qualquer Servidor OPC, independentemente do tipo de dispositivo e do fabricante. Essa comunicação é válida

somente para OPC-DA (*Data Access*), uma vez que existem diferentes tecnologias OPC.

Basicamente, o padrão OPC estabelece as regras para que sejam desenvolvidos sistemas com interfaces padrões para comunicação dos dispositivos de campo (controladores, sensores *etc.*) com sistemas de monitoração, supervisão e gerenciamento (SCADA, MES, ERP *etc.*) (Duarte *et al*, 2006). No Apêndice B é mostrado em detalhes o código desenvolvido em Matlab para acesso ao servidor OPC.

### **3.2.2 Módulo de Inicialização**

Na etapa de inicialização, configuram-se primeiramente o tempo de amostragem, o tempo total de simulação e os parâmetros de sintonia do controlador preditivo (horizonte de predição, horizonte de controle, pesos *etc.*). Realiza-se a leitura dos dados dos instrumentos da planta via *OPC*, constrói-se o modelo linear do processo para ser utilizado pelo controlador e inicializam-se os parâmetros do modelo bem como as condições iniciais. Todas as variáveis, parâmetros e objetos inicializados neste módulo são disponibilizados no *workspace* para serem acessados pelos demais módulos da estrutura desenvolvidos em *MATLAB*.

### **3.2.3 Módulo de Simulação do Processo**

Para simulação da planta foi utilizado o *software* EMSO (*Environment for Modeling, Simulation and Optimization*) (Soares e Secchi, 2003).

O *software* EMSO é um ambiente integrado de ferramentas computacionais para a síntese e análise de processos industriais químicos e petroquímicos, visando atender às necessidades de um simulador nacional para ensino e treinamento, projeto e otimização de processos. O EMSO possui desenvolvimento continuado através do projeto ALSOC da UFRGS, COPPE e USP. No seu estágio atual de desenvolvimento, este simulador é capaz de realizar as seguintes tarefas:

- ✓ Simulações estacionárias
- ✓ Simulações dinâmicas
- ✓ Otimização estacionária (NLP, MINLP)
- ✓ Estimção de parâmetros com modelos estacionários
- ✓ Estimção de parâmetros com modelos dinâmicos
- ✓ Reconciliação de dados estacionária
- ✓ Acompanhamento de processos e inferências com comunicação OPC
- ✓ Construção de diagramas de bifurcação (interface com AUTO para DAEs)
- ✓ Simulação dinâmica com o SIMULINK (interface com MATLAB)
- ✓ Adição de novos *solvers* (DAE, NLA, NLP)
- ✓ Adição de rotinas externas usando o recurso de *Plugins*

A sua capacidade de tratar com sistemas de equações algébrico-diferenciais de índice elevado (Brenan e Petzold, 1989; Costa *et al.*, 2003) é um grande diferencial deste simulador e importante para a solução de problemas de D-RTO.

Neste trabalho, o EMSO será usado para solução do sistema de equações algébrico-diferenciais funcionando como um simulador da planta do processo.

### **3.2.4 Módulo de Otimização Dinâmica**

O módulo de Otimização Dinâmica é responsável por fornecer a trajetória de referência ótima para o controlador. Para isso, é necessário formular o problema de otimização dinâmica e construir o modelo do processo.

Na etapa de formulação do problema de otimização dinâmica, configura-se o problema de otimização onde se estabelece a função objetivo, constrói-se o modelo do processo e se estabelecem as restrições do problema. Na etapa de solução do problema de otimização dinâmica é executado o tratamento do problema de otimização (ex.: discretização das variáveis). Uma vez que o problema esteja apresentado na forma adequada, um algoritmo de otimização resolve o problema de otimização dinâmica, calculando a trajetória ótima das variáveis de controle. Uma vez resolvido o problema, as trajetórias calculadas devem ser validadas e implementadas nas aplicações de MPC, na forma de trajetórias de referência para as variáveis de controle, ou diretamente nos instrumentos da planta, na forma de *setpoints* para o controle regulatório.

Neste módulo, será implementado um método direto com *single-shooting* para resolver o problema de otimização dinâmica, que consiste da aproximação da trajetória

de controle por uma função paramétrica com forma funcional definida, deixando as equações de estado na forma original do sistema de DAE.

Esta aproximação é feita dividindo o horizonte de tempo do problema de otimização em  $NE$  subintervalos onde as variáveis de controle são representadas por funções que podem ser constantes por partes, lineares por partes ou por uma aproximação polinomial. Com a escolha da forma dos perfis das variáveis de controle, a função original se transforma em  $\tilde{F}(\dot{x}, x, \hat{u}) = 0$ , onde  $\hat{u}$  são os novos parâmetros de otimização, invariantes no tempo. Neste caso, o problema passa a ter apenas os parâmetros de otimização ( $\hat{u}$ ) e a função objetivo. Este método é do tipo caminho viável (*feasible path*), isto é, a solução é melhorada a cada iteração.

Quando o método de otimização faz uso do gradiente da função objetivo, o método *single-shooting* necessita que as equações de estado e de sensibilidade sejam integradas simultaneamente para dados perfis de controle. Com isso, fornece os valores das funções e gradientes para o algoritmo de NLP. O algoritmo de otimização é aplicado no laço externo para atualizar as ações de controle.

As equações de sensibilidade são obtidas, de forma eficiente, por diferenciação das equações de processo, após a discretização das variáveis de controle, em relação ao conjunto de parâmetros. Dada a sensibilidade dos estados em relação aos parâmetros, as derivadas da função objetivo e das restrições em relação aos parâmetros são facilmente calculadas.

## **Procedimento**

A forma geral do método *single-shooting* direto é descrito a seguir (vide Figura 3.2.4.1).

Seja o problema de otimização dinâmica dado por:

$$\begin{aligned}
 & \min_{x(t), u(t), t_f, p} J(x(t), u(t), t, p) = \phi(x(t_f), t_f) \\
 & \text{sujeito a :} \\
 & F(\dot{x}(t), x(t), u(t), t, p) = 0 \\
 & G(x(t), u(t), t, p) \leq 0, \quad \forall t \in T \\
 & I(x_0, u_0, t_0, p) = 0 \\
 & x_{\min} \leq x(t) \leq x_{\max} \\
 & u_{\min} \leq u(t) \leq u_{\max}
 \end{aligned} \tag{3.2.4.1}$$

Passo 1 – Definir uma função de parametrização das variáveis de controle (entradas do processo).

$$u_i(t) = \sum_{k=1}^{NE} \hat{u}_{i,k} \psi_k(t) \quad i = 1, \dots, nu \tag{3.2.4.2}$$

Passo 2 – Reformular o problema de otimização incluindo os parâmetros de controle, na forma:

$$\begin{aligned}
 & \min_{\hat{u}} J = \phi(x(t_f), t_f) \\
 & \text{sujeito a :} \\
 & F(\dot{x}(t), x(t), u(t), t, p) = 0 \\
 & G(x(t), u(t), t, p) \leq 0 \\
 & u_i(t) = \sum_{k=1}^{NE} \hat{u}_{i,k} \psi_k(t) \quad i = 1, \dots, nu \\
 & I(x_0, u_0, t_0, p) = 0 \\
 & x_{\min} \leq x(t) \leq x_{\max} \\
 & u_{\min} \leq u(t) \leq u_{\max}
 \end{aligned} \tag{3.2.4.3}$$

Passo 3 – Escolher estimativas iniciais dos parâmetros  $\hat{u}$ .

Passo 4 – Integrar o modelo dinâmico dado por  $F(\dot{x}(t), x(t), u(t), t, p) = 0$  e  $I(x_0, u_0, t_0, p) = 0$  para todas as entradas de  $t_0$  a  $t_f$ , usando algum *solver* de DAE (ex.: DASSLC).

Passo 5 – Avaliar a função objetivo  $J$  e as restrições  $G(x(t), u(t), t, p) \leq 0$  usando os perfis de estado obtido no passo 4 e os parâmetros de controle  $u(t)$  proposto.

Passo 6 – Atualizar os valores das variáveis de decisão  $\hat{u}$  com algum método de NLP.

Passo 7 – Verificar a convergência de  $\hat{u}$ . Caso seja satisfeita, a otimização está concluída, caso contrário repetir os passos 4 a 7 até convergir.

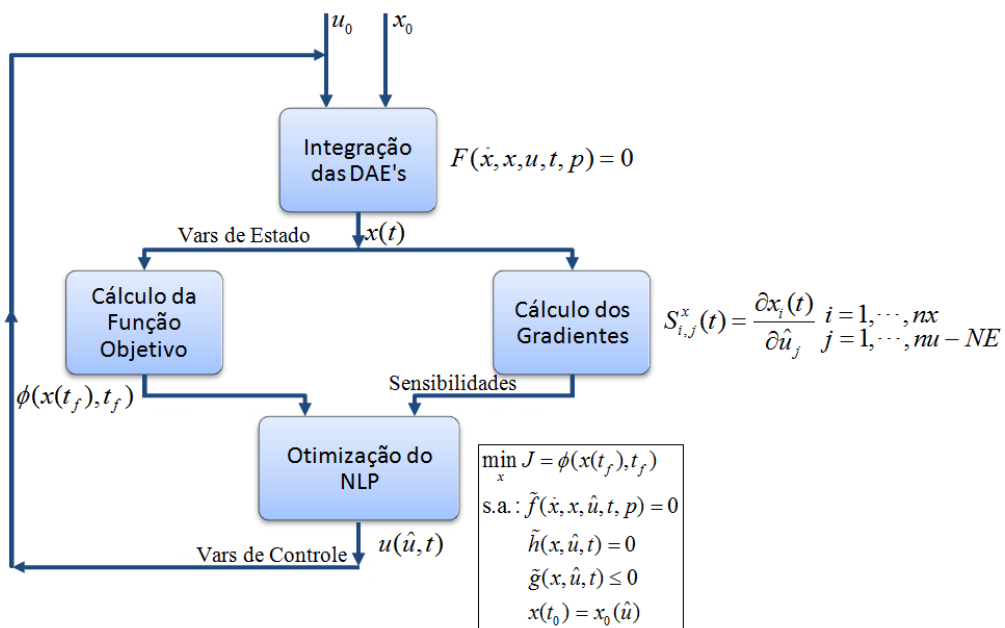


Figura 3.2.4.1 – Método *single-shooting* direto.



Para usar um algoritmo de NLP, é necessário calcular as derivadas das restrições de desigualdade  $G(x(t), u(t), t, p) \leq 0$ . Estes cálculos envolvem as derivadas parciais  $\partial x / \partial \hat{u}$  que são soluções das equações diferenciais de sensibilidade. Este sistema deve ser resolvido junto com o conjunto original de DAEs.

### **Cálculo dos Gradientes**

Os gradientes podem ser calculados de três formas: por perturbações nos parâmetros, pelas equações de sensibilidade, ou pelas equações adjuntas.

#### ***Por Perturbações nos Parâmetros***

Calculando o gradiente aproximado da função objetivo pela solução do modelo DAE ( $nu \cdot NE + 1$ ) vezes e fazendo:

$$\frac{d\phi}{d\hat{u}_i} = \frac{\phi(\hat{u}_i + \Delta\hat{u}_i)}{\Delta\hat{u}_i} \quad (3.2.4.4)$$

Sendo que o desempenho do otimizador e a eficiência para encontrar a solução ótima estão diretamente ligados à qualidade do gradiente produzido a cada iteração. O esforço computacional é proporcional ao número de variáveis de entrada.

#### ***Pelas Equações de Sensibilidade***

A derivada da função objetivo em relação aos parâmetros é determinada da seguinte maneira.

Seja o problema de otimização definido por:

$$\begin{aligned}
 & \min_{\hat{u}} J = \phi(x(\hat{u}, t_f)) \\
 & \text{s.a. :} \\
 & \quad \tilde{f}(\dot{x}, x, \hat{u}) = 0 \\
 & \quad \tilde{h}(x, \hat{u}) = 0 \\
 & \quad \tilde{g}(x, \hat{u}) \leq 0 \\
 & \quad x(t_0) = x_0(\hat{u})
 \end{aligned} \tag{3.2.4.5}$$

A matriz de sensibilidade do sistema DAE é definida como:

$$\frac{\partial x}{\partial \hat{u}} \quad \text{e} \quad \frac{\partial x_0}{\partial \hat{u}}$$

A solução das equações de sensibilidade é obtida a partir da jacobiana do sistema DAE, que é calculada a cada passo de integração. O esforço computacional é reduzido a uma multiplicação de matrizes por parâmetro por avaliação de jacobiano. Os gradientes da função objetivo e das restrições podem ser calculados usando a regra da cadeia:

$$\begin{aligned}
 \frac{\partial J}{\partial \hat{u}_j} &= \sum_{i=1}^{nx} \frac{\partial \phi}{\partial x_i} \frac{\partial x_i}{\partial \hat{u}_j} \Bigg|_{t=t_f} \quad \text{onde} \quad s_{i,j}^x = \frac{\partial x_i}{\partial \hat{u}_j} \\
 \text{ou} \\
 \frac{\partial J}{\partial \hat{u}} &= \left( \frac{\partial \phi}{\partial x} \right) S^x \Bigg|_{t=t_f} \\
 \text{onde} \quad S^x &= \left( \frac{\partial x}{\partial \hat{u}} \right)
 \end{aligned} \tag{3.2.4.6}$$

Diferenciando as equações de sensibilidade em relação ao tempo, obtém-se

$$\dot{S}^x = \frac{ds^x}{dt} = \frac{d}{dt} \left( \frac{\partial x}{\partial \hat{u}} \right) = \frac{\partial \dot{x}}{\partial \hat{u}} \quad (3.2.4.7)$$

Usando a regra da cadeia na diferenciação do sistema de DAEs, obtém-se:

$$\begin{aligned} \frac{\partial \tilde{f}}{\partial \dot{x}} \frac{ds_j^x}{dt} + \frac{\partial \tilde{f}}{\partial x} s_j^x + \frac{\partial \tilde{f}}{\partial u_j} &= 0 \\ S_j^x(0) &= \frac{\partial x(0)}{\partial \hat{u}_j} \end{aligned} \quad (3.2.4.8)$$

Estas equações são integradas junto com as equações de estado por meio de *solvers* de DAE como DASSLC ou PSIDE, obtendo-se os valores de  $x(t_f)$  e  $s(t_f)$ .

Apesar dos avanços na solução das equações de sensibilidade, o esforço computacional continua a ser um ponto importante nos algoritmos de otimização. O custo computacional está diretamente ligado ao número de variáveis de entrada.

### ***Pelas Equações Adjuntas***

A abordagem adjunta ou dual para sensibilidade é dada pelo seguinte procedimento. Seja o problema de otimização definido por:

$$\begin{aligned} \min_{\hat{u}} J &= \phi(x(\hat{u}, t_f)) \\ \text{s.a. :} \\ \dot{x} &= \tilde{f}(x, \hat{u}, t) \\ \tilde{h}(x, \hat{u}, t) &= 0 \\ \tilde{g}(x, \hat{u}, t) &\leq 0 \\ x(t_0) &= x_0(\hat{u}) \end{aligned} \quad (3.2.4.9)$$

Acrescentando o modelo à função objetivo

$$\Phi = \phi(x(\hat{u}, t_f)) - \int_0^{t_f} \lambda^T (\dot{x} - \tilde{f}(x, y, \hat{u}, t)) dt \quad (3.2.4.10)$$

onde  $\lambda$  é o multiplicador do sistema de EDOs e integrando  $\Phi$  por partes, tem-se:

$$\Phi = \phi(x(\hat{u}, t_f)) + \lambda(0)^T x_0(\hat{u}) - \lambda(t_f)^T x(\hat{u}, t_f) + \int_0^{t_f} (x^T \dot{\lambda} + \lambda^T f(x, \hat{u}, t)) dt$$

e

$$\begin{aligned} \delta\Phi = & \left( \frac{\partial\phi(x(\hat{u}, t_f))}{\partial x(\hat{u}, t_f)} - \lambda(t_f)^T \right) \delta x(\hat{u}, t_f) + \left[ \frac{\partial x_0(\hat{u})}{\partial \hat{u}} \lambda(0) \right]^T \delta \hat{u} + \\ & + \int_0^{t_f} \left( \left[ \dot{\lambda} + \frac{\partial \tilde{f}}{\partial x} \lambda \right]^T \delta x + \left[ \frac{\partial \tilde{f}}{\partial \hat{u}} \lambda \right] \delta \hat{u} \right) dt \end{aligned} \quad (3.2.4.11)$$

Como todos os termos relacionados à  $\delta x$  são nulos, tem-se:

$$\begin{aligned} \frac{\partial\phi(x(\hat{u}, t_f))}{\partial x(\hat{u}, t_f)} - \lambda(t_f)^T = 0 \quad \text{ou} \quad \lambda(t_f) &= \frac{\partial\phi(x(\hat{u}, t_f))}{\partial x(\hat{u}, t_f)} \\ \dot{\lambda} + \frac{\partial \tilde{f}}{\partial x} \lambda = 0 \quad \text{ou} \quad \dot{\lambda}(t) &= -\frac{\partial \tilde{f}}{\partial x} \lambda(t) \end{aligned} \quad (3.2.4.12)$$

$$\frac{d\phi}{d\hat{u}} = \frac{\partial x_0(\hat{u})}{\partial \hat{u}} \lambda(0) + \int_0^{t_f} \left( \frac{\partial \tilde{f}}{\partial \hat{u}} \lambda(t) \right) dt$$

Deve-se integrar o modelo  $\tilde{f}(x, \hat{u}, t)$  para frente e as equações adjuntas  $\dot{\lambda}(t)$  para trás e avaliar a integral e a sensibilidade.

A abordagem sequencial tende a ser lenta pelo fato que o método *feasible path* requer repetidas e custosas soluções das equações diferenciais. A parte mais custosa é a etapa da integração precisa das equações dinâmicas, mesmo quando as variáveis de decisão estão longe do ponto ótimo.

A qualidade da solução está diretamente ligada à parametrização das variáveis de controle. A solução converge rapidamente somente se a estimativa inicial for uma solução viável. Entretanto, encontrar uma solução viável não é um problema trivial.

### **3.2.5 Módulo de Controle Avançado**

Neste módulo, o algoritmo MPC emprega um modelo linear do processo para prever seu comportamento futuro sobre um horizonte de predição estendido. Um objetivo de desempenho a ser minimizado é definido sobre o horizonte de predição, usualmente uma soma dos quadrados dos erros em relação ao *setpoint* e das ações de controle. Esta função custo é minimizada avaliando um perfil de movimentos da entrada manipulada a ser implementado em instantes de amostragens sucessivos sobre o horizonte de controle.

Os diversos algoritmos MPC propõem diferentes funções objetivo para obtenção da lei de controle. Uma expressão geral da função objetivo é,

$$J = \sum_{j=P_0}^P \|\hat{y}(t+j|t) - r(t+j)\|_Q^2 + \sum_{j=1}^M \|\Delta u(t+j-1)\|_w^2 \quad (3.2.5.1)$$

onde  $\hat{y}(t+j|t)$  é a saída predita  $j$  passos no futuro baseado em informação disponível no tempo  $t$ ,  $r(t+j)$  é o sinal referência  $j$  passos no futuro,  $\Delta u(t)=u(t)-u(t-1)$ , e  $\|x\|_W^2$  é a norma Euclidiana ponderada de  $x \in \mathfrak{R}^n$  definida como  $\|x\|_W^2 = x^T W x$  com  $W \in \mathfrak{R}^{n \times n}$  positiva-definida.

Os parâmetros de sintonia são: início do horizonte de predição ( $P_0$ ), término do horizonte de predição ( $P$ ), horizonte de controle ( $M$ ), tempo de amostragem ( $t_s$ ), peso da variável controlada ( $Q$ ) e coeficiente de supressão de movimentos ( $W$ ). As matrizes de peso  $Q$  e  $W$  podem ser escolhidas como variantes no tempo (i.e., funções de  $j$ ). Neste trabalho, por simplicidade, elas são consideradas invariantes no tempo.

Cada um dos parâmetros acima tem regras específicas de sintonia em algoritmos MPC. Usando a experiência acumulada de aplicação de algoritmos preditivos, algumas regras de engenharia têm sido identificadas para obter valores apropriados dos parâmetros para bom desempenho em diferentes aplicações (Trierweiler e Farina, 2003).

Neste trabalho foi utilizado o *Model Predictive Control Toolbox* do Matlab para solução do problema de otimização do controlador, onde a solução ótima é computada usando o método de programação quadrática (*QP solver*).

### **3.2.6 Módulo Disparador do D-RTO**

O módulo Disparador do D-RTO consiste de um algoritmo contendo critérios para ativação do D-RTO. Este algoritmo é executado em tempo de simulação (*on-line*)

para atualização das trajetórias de referência das variáveis de otimização no controlador, permitindo assim que o módulo de otimização seja executado somente em situações necessárias. Cada vez que o D-RTO é executado, é gerado um *log* contendo o tempo da execução e o motivo. Os critérios utilizados pelo algoritmo para execução do D-RTO estão descritos abaixo. O disparador irá executar o D-RTO caso os critérios de ciclo e *setpoint* sejam verdadeiros ou quando o critério de perturbação seja verdadeiro, caso contrário, será feita apenas a execução do MPC.

- Critério de Ciclo: Será verdadeiro cada vez que o tempo de simulação ( $k$ ) for múltiplo do tempo de ciclo ( $t_c$ ) definido pelo usuário;

$$k = i \cdot t_c, \quad i \in \mathfrak{R}^n$$

- Critério de Setpoint: Verifica se a variável controlada ( $y_i$ ) está no *setpoint*. Será verdadeiro se o valor corrente da variável de otimização ( $x_{i,k}$ ) for igual ao último *setpoint* calculado ( $SP_{i,k}$ ) mais ou menos um determinado *offset* ( $\Delta$ );

$$Sp_{i,k} + \Delta \leq x_{i,k} \leq SP_{i,k} - \Delta$$

- Critério de Perturbação: Será verdadeiro quando ocorrer uma perturbação, ou seja, quando o valor da variável de entrada medida corrente for diferente do seu valor na iteração anterior.

$$u_{i,k} \neq u_{i,k-1}$$

- Critério Específico: Este critério é utilizado em situações específicas durante um período de tempo definido, como por exemplo durante um *start-up*. Nele podem ser definidas programações de execução do D-RTO, como por exemplo em tempos de ciclo menores que o definido no critério de ciclo, bem como outras lógicas customizadas para cada caso. Durante o período que este critério está ativo, os demais critérios devem ser desativados para não haver conflito.



## 4 Estudo de Caso: Cinética de Van der Vusse

O estudo de caso no presente trabalho consiste na aplicação do sistema de otimização dinâmica em um reator CSTR com a cinética de Van der Vusse. O esquema de reação de Van der Vusse tem sido frequentemente usada como um problema *benchmark* para algoritmos de controle de processo não-lineares (Engell e Klatt, 1993; Chen *et al.*, 1995). Este modelo apresenta comportamento não-linear dependendo dos pontos de operação considerados e é útil do ponto de vista da análise de desempenho da estratégia proposta. A complexidade relativamente baixa permite seu uso para comparação, evitando assim efeitos de distorção que resultariam da redução do modelo. Além do mais, este modelo é interessante do ponto de vista da aplicação industrial, pois representa uma classe de CSTR com resfriamento, como descrito na Figura 4.1.1.

Com objetivo de testar a estrutura proposta foram realizados os seguintes estudos:

1<sup>a</sup> – Avaliação do Módulo D-RTO

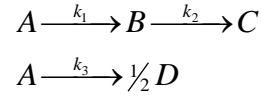
2<sup>a</sup> – Avaliação do Módulo Disparador D-RTO

Os resultados destes estudos serão apresentados e discutidos no Capítulo 5.

### 4.1 Descrição do Processo

O reator é alimentado com uma solução de um reagente A com concentração  $C_{Af}$ . As reações químicas no reator seguem o mecanismo de Van der Vusse, onde a

reação principal  $A \rightarrow B$  é acompanhada por uma reação paralela  $A \rightarrow \frac{1}{2} D$  e uma reação em série  $B \rightarrow C$ :



O componente  $B$  é o produto desejado, e  $C$  e  $D$  são subprodutos não desejados. A reação de Van der Vusse é realizada em um reator contínuo de tanque agitado (CSTR) não-isotérmico. O modelo do reator, considerando volume e propriedades físicas constantes, compreende o balanço de massa para cada componente e o balanço de energia:

$$\begin{aligned} \frac{d}{dt} C_A(t) &= \frac{C_{Af} - C_A(t)}{\tau} - r_1 - r_3 \\ \frac{d}{dt} C_B(t) &= \frac{C_{Bf} - C_B(t)}{\tau} + r_1 - r_2 \\ \frac{d}{dt} C_C(t) &= \frac{C_{Cf} - C_C(t)}{\tau} + r_2 \\ \frac{d}{dt} C_D(t) &= \frac{C_{Df} - C_D(t)}{\tau} + \frac{1}{2} r_3 \\ \frac{d}{dt} T(t) &= \frac{T_f - T(t)}{\tau} + \frac{q_r - q}{\rho V c_p} \end{aligned} \quad (4.1.1)$$

Nestas equações  $C_{if}$ ,  $i = A, B, C$  ou  $D$  são as concentrações dos componentes na corrente de alimentação;

$\tau = V/F_f$  é o tempo de residência;

$F_f$  é a vazão de alimentação;

$V$  é o volume do reator;

$c_p$  é o calor específico;

$r_i$ ,  $i = 1, 2$  ou  $3$ , são as taxas de reação dadas por:

$$\begin{aligned}r_1 &= k_1 C_A(t) \\r_2 &= k_2 C_B(t) \\r_3 &= k_3 C_A^2(t)\end{aligned}\tag{4.1.2}$$

As cinéticas de reação são modeladas usando expressões de Arrhenius:<sup>2</sup>

$$k_i(T) = k_{i0} \exp\left(\frac{-E_i}{T}\right), \quad i = 1, 2 \text{ ou } 3.\tag{4.1.3}$$

A taxa de calor de reação  $q_r$  e a taxa de transferência de calor  $q$  são dadas por:

$$\begin{aligned}q_r &= -(\Delta H_{r1} r_1 + \Delta H_{r2} r_2 + \Delta H_{r3} r_3)V \\q &= U A_t [T(t) - T_w]\end{aligned}\tag{4.1.4}$$

onde:

$\Delta H_{ri}$  é o calor de reação;

$U$  é o coeficiente global de transferência de calor;

$A_t$  é a área de transferência de calor.

---

<sup>2</sup> Os parâmetros da energia de ativação  $E_i$  já compreendem a constante de gás e então têm unidades de K.

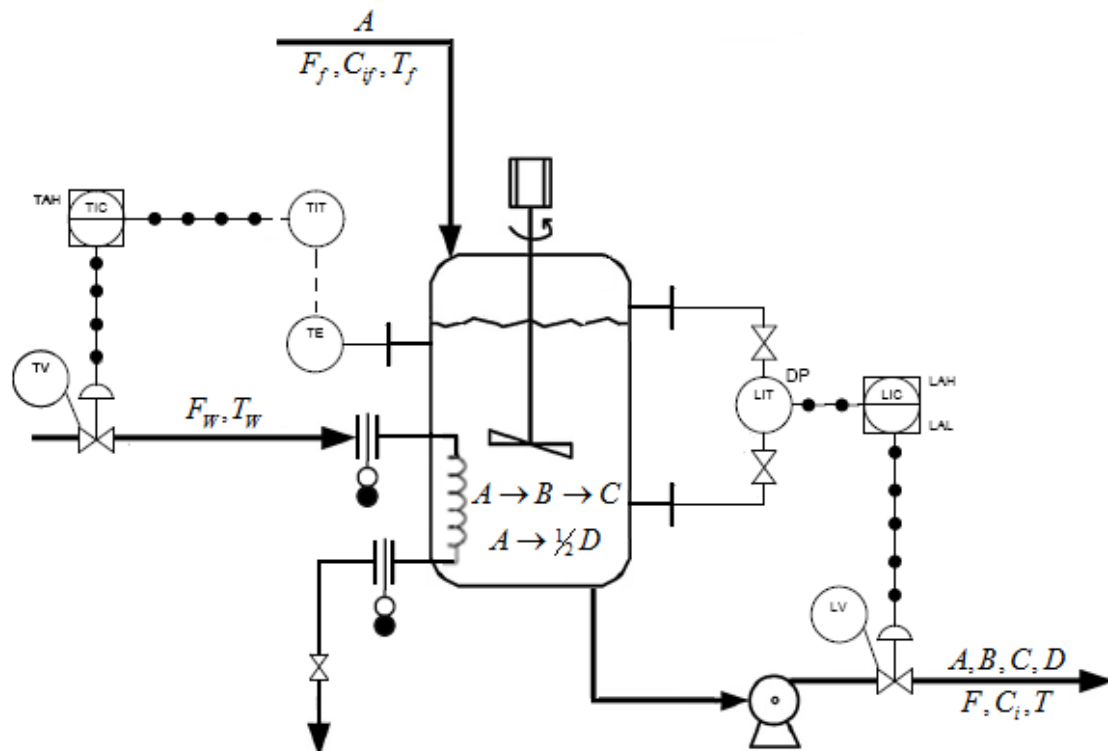


Figura 4.1.1 – Esboço de um CSTR com resfriamento e reação de Van der Vusse.

As concentrações e temperaturas constituem o estado  $x=[C_A \ C_B \ C_C \ C_D \ T]^T$  do sistema dinâmico não-linear (4.1.1). Os parâmetros do modelo estão relacionados na Tabela 4.1.1. No esquema representado na Figura 4.1.1, foi considerado controle perfeito do nível (volume constante) e atuação direta na temperatura da camisa através da malha de controle da temperatura do reator.

Tabela 4.1.1 – Parâmetros do Modelo

$\Delta H_1 = 4200 \text{ kJ/kmol}$	$E_1 = 9758,3 \text{ K}$
$\Delta H_2 = -11000 \text{ kJ/kmol}$	$E_2 = 9758,3 \text{ K}$
$\Delta H_3 = -41850 \text{ kJ/kmol}$	$E_3 = 8560 \text{ K}$
$c_p = 2,8121 \text{ kJ/kg/K}$	$D_i = 2 \text{ m}$
$\rho = 1000 \text{ kg/m}^3$	$H = 4 \text{ m}$
$k_1 = 1,287 \times 10^{12} \text{ l/min}$	$U = 642,18 \text{ kJ/min/m}^2/\text{K}$
$k_2 = 1,287 \times 10^{12} \text{ l/min}$	
$k_3 = 9,043 \times 10^9 \text{ l/mol/min}$	

A vazão de entrada  $F_f$  e a temperatura de resfriamento  $T_w$  servem como variáveis manipuladas.

$$u_1 = F_f, \quad u_2 = T_w \quad (4.1.5)$$

Ambas as entradas são sujeitas a restrições na forma de limites inferior e superior:

$$\begin{aligned} u_1^- = 1 \text{ l/min} \leq u_1 \leq 50 \text{ l/min} = u_1^+, \\ u_2^- = 300K \leq u_2 \leq 400K = u_2^+ \end{aligned} \quad (4.1.6)$$

A concentração do produto  $C_B$  e a temperatura do reator  $T$  são uma escolha natural para a saída controlada:

$$y_1 = C_B, \quad y_2 = T \quad (4.1.7)$$

## 4.2 Dinâmica do Sistema

O gráfico da concentração de  $C_B$  no estado estacionário contra  $F_f$  para uma dada temperatura do reator,  $T$ , e diferentes composições de entrada  $C_{Af}$  (Figura 4.2.1) revela um comportamento interessante do sistema. O reator exibe uma mudança no ganho no pico da produção do reator (i.e., onde a concentração de  $C_B$  alcança seu valor máximo), e apresenta comportamento de fase não-mínima (resposta inversa) para pontos de operação à esquerda deste pico e comportamento de fase mínima com *overshoot* para pontos de operação à direita. As três curvas mostradas na Figura 4.2.1 foram obtidas

para três diferentes valores de  $C_{Af}$ , a faixa de variação da vazão utilizada foi de 0 a 50 l/min, porém, para melhorar a visualização, no gráfico é mostrado até 7 l/min.

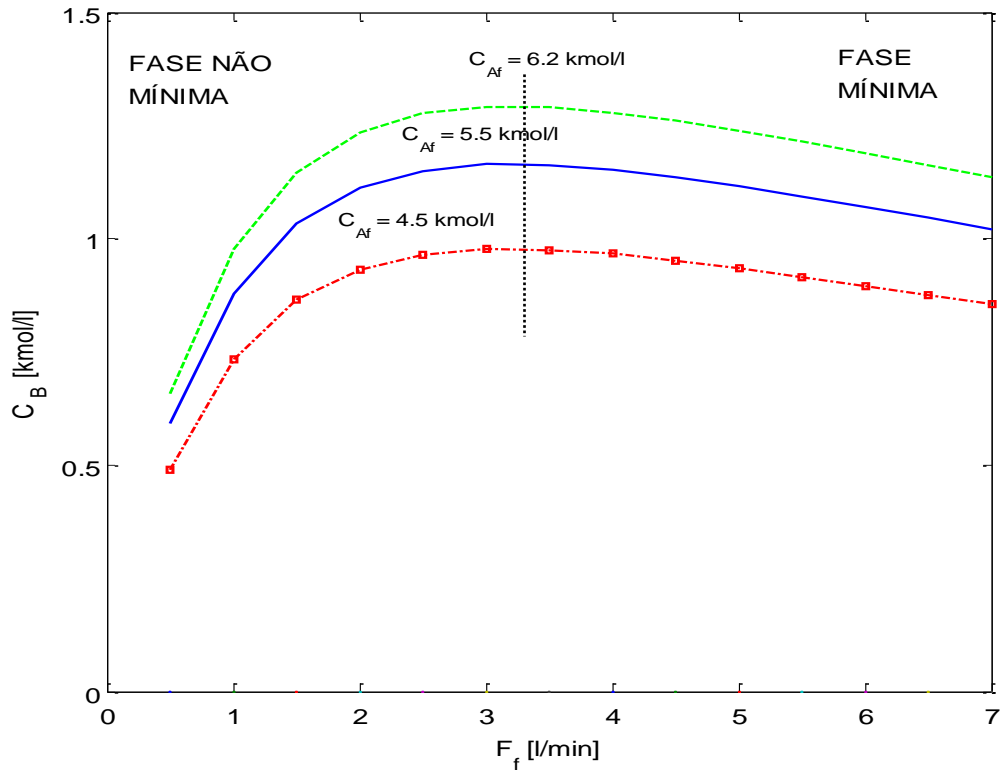


Figura 4.2.1 –  $C_B$  como função de  $F_f$  para  $T = 378,05K$  e diferentes concentrações de  $C_{Af}$ .

De forma a evitar o comportamento da resposta inversa, trabalhou-se, neste estudo de caso, na região de fase mínima.

### 4.3 Construção do Problema de Otimização Dinâmica

Conforme apresentado anteriormente em (3.1.1.1.1), o problema de otimização dinâmica de um processo tem a seguinte formulação geral:

$$\min_{x(t), y(t), t_f, p} \varphi(x(t), y(t), u(t))$$

sujeito a :

$$\begin{aligned} F\left(\frac{dx(t)}{dt}, x(t), y(t), u(t)\right) &= 0 && \text{Modelo Dinâmico} \\ G(x(t), y(t), u(t)) &= 0 && \text{Equações Constitutivas (EA)} \\ x(0) &= x^0 && \text{Condições Iniciais} \\ x^- &\leq x(t) \leq x^+ && \\ y^- &\leq y(t) \leq y^+ && \text{Limites} \\ u^- &\leq u(t) \leq u^+ && \end{aligned} \quad (4.3.1)$$

### 4.3.1 Formulação do DAOP

O objetivo de produção neste estudo é maximizar a concentração do componente *B* no tempo final, que é o produto desejado, reduzindo a produção dos componente não-desejados *C* e *D*. Reescrevendo o problema geral dado pela equação (4.3.1) para o modelo do reator CSTR descrito na equação (4.1.1), tem-se:

$$\max_{F_f, T_w, t_f} C_B(t_f)$$

sujeito a :

$$\begin{aligned} \frac{d}{dt} C_A(t) &= \frac{C_{Af} - C_A(t)}{\tau} - r_1 - r_3 \\ \frac{d}{dt} C_B(t) &= \frac{C_{Bf} - C_B(t)}{\tau} + r_1 - r_2 \\ \frac{d}{dt} C_C(t) &= \frac{C_{Cf} - C_C(t)}{\tau} + r_2 \\ \frac{d}{dt} C_D(t) &= \frac{C_{Df} - C_D(t)}{\tau} + \frac{1}{2} r_3 \\ \frac{d}{dt} T(t) &= \frac{T_f - T(t)}{\tau} + \frac{q_r - q}{\rho V c_p} \end{aligned}$$

$$C_i(t) \geq 0 \quad i=A, B, C, D.$$

$$C_i(0) = C_{i0}$$

$$F_{f \min} \leq F_f \leq F_{f \max}$$

$$T_{W \min} \leq T_W \leq T_{W \max} \quad (4.3.1.1)$$

$$\begin{aligned}
r_1 &= k_1 C_A(t) \\
r_2 &= k_2 C_B(t) \\
r_3 &= k_3 C_A^2(t)
\end{aligned}$$

$$\begin{aligned}
q_r &= -(\Delta H_{r1} r_1 + \Delta H_{r2} r_2 + \Delta H_{r3} r_3) V \\
q &= U A_i [T(t) - T_w]
\end{aligned}$$

$$k_i(T) = k_{i0} \exp\left(\frac{-E_i}{T}\right), \quad i = 1, 2 \text{ ou } 3.$$

### 4.3.2 Solução do Problema de Otimização Dinâmica

Para solução do problema de otimização dinâmica dado pela equação (4.3.1.1), foi implementado o método *single-shooting* direto cujo algoritmo de solução foi apresentado no Capítulo 3 (vide Figura 3.2.4.1). Este método consiste da aproximação da trajetória de controle por uma função paramétrica com forma funcional conhecida, deixando as equações de estado na forma original do sistema DAE. Esta aproximação é feita dividindo o horizonte de tempo do problema de otimização em  $NE$  subintervalos onde as variáveis de controle foram representadas neste trabalho por funções constantes por partes:

$$u_i(t) = \sum_{k=1}^{NE} \hat{u}_{i,k} \psi_k(t) \quad \text{onde} \quad \psi_k(t) = \begin{cases} 1 & \forall t \in [t_k, t_{k+1}] \\ 0 & \forall t \notin [t_k, t_{k+1}] \end{cases} \quad (4.3.2.1)$$

onde  $\hat{u}_{i,k}$  é o valor da variável de controle  $i$  no subintervalo  $k$ .



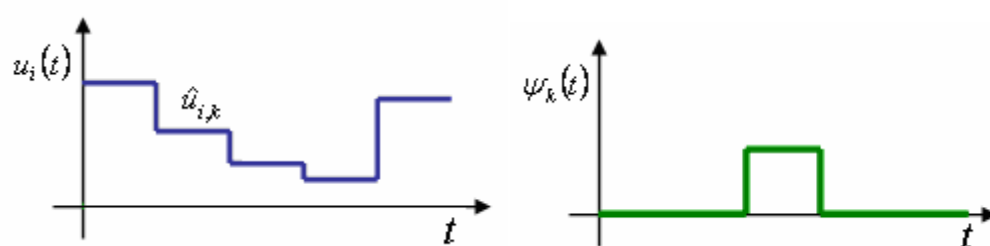


Figura 4.3.2.1 – Funções constantes por partes.

Desta forma, o problema de otimização dado pela equação (4.3.1.1) pode ser reformulado incluindo os parâmetros de controle da seguinte maneira:

$$\begin{aligned}
 & \max_{\hat{u}} C_B(t_f) \\
 & \text{sujeito a :} \\
 & \frac{d}{dt} C_A(t) = \frac{C_{Af} - C_A(t)}{\tau} - r_1 - r_3 \\
 & \frac{d}{dt} C_B(t) = \frac{C_{Bf} - C_B(t)}{\tau} + r_1 - r_2 \\
 & \frac{d}{dt} C_C(t) = \frac{C_{Cf} - C_C(t)}{\tau} + r_2 \\
 & \frac{d}{dt} C_D(t) = \frac{C_{Df} - C_D(t)}{\tau} + \frac{1}{2} r_3 \\
 & \frac{d}{dt} T(t) = \frac{T_f - T(t)}{\tau} + \frac{q_r - q}{\rho V c_p} \\
 & u_i(t) = \sum_{k=1}^{NE} \hat{u}_{i,k} \psi_k(t) \quad i=1,2. \\
 & C_i(0) = C_{i0} \\
 & C_i(t) \geq 0 \quad i=A, B, C, D. \\
 & u_{\min} \leq u_k(t) \leq u_{\max}
 \end{aligned} \tag{4.3.2.2}$$

onde  $\hat{u}$  são os parâmetros de otimização invariantes no tempo e  $u(t) = [F_f(t) \quad T_w(t)]$  e  $t_f$  é o horizonte do problema de otimização dinâmica.

O modelo dinâmico é integrado para todas as entradas entre  $t_0$  e  $t_f$  utilizando o *solver* de ODEs *ode15s* do MATLAB. O problema NLP é resolvido por um algoritmo de programação quadrática sequencial (SQP), implementado no *solver* *fmincon* do MATLAB. Neste caso, o método *single-shooting* necessita ainda que as equações de estado e de sensibilidade sejam integradas simultaneamente para o perfil de controle dado. Com isso, são fornecidos os valores das funções e gradientes para o algoritmo de NLP.

### Cálculo das Equações de Sensibilidade

Para usar o algoritmo de NLP é necessário conhecer a influência das variáveis de decisão nas variáveis de otimização realizando uma análise de sensibilidade a partir do sistema dinâmico descrito pela equação (4.3.2.2).

Conforme apresentado na equação (3.2.4.8), as equações de sensibilidade são obtidas usando a regra da cadeia na diferenciação do sistema de DAEs,

$$\dot{W}(t) = J_x W(t) + J_u \quad (4.3.2.3)$$

onde  $W(t)$  é a matriz de sensibilidade das variáveis de estado em relação às entradas do sistema, e  $J_x$  e  $J_u$  são as matrizes jacobianas do sistema.

Para o sistema em estudo, tem-se:

$$W(t) = \frac{\partial x(t)}{\partial u} = \begin{bmatrix} \frac{\partial C_A}{\partial F_f} & \frac{\partial C_A}{\partial T_w} \\ \frac{\partial C_B}{\partial F_f} & \frac{\partial C_B}{\partial T_w} \\ \frac{\partial C_C}{\partial F_f} & \frac{\partial C_C}{\partial T_w} \\ \frac{\partial C_D}{\partial F_f} & \frac{\partial C_D}{\partial T_w} \\ \frac{\partial T}{\partial F_f} & \frac{\partial T}{\partial T_w} \end{bmatrix}$$

$$J_x = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial C_A} & \frac{\partial f_1}{\partial C_B} & \frac{\partial f_1}{\partial C_C} & \frac{\partial f_1}{\partial C_D} & \frac{\partial f_1}{\partial T} \\ \frac{\partial f_2}{\partial C_A} & \frac{\partial f_2}{\partial C_B} & \frac{\partial f_2}{\partial C_C} & \frac{\partial f_2}{\partial C_D} & \frac{\partial f_2}{\partial T} \\ \frac{\partial f_3}{\partial C_A} & \frac{\partial f_3}{\partial C_B} & \frac{\partial f_3}{\partial C_C} & \frac{\partial f_3}{\partial C_D} & \frac{\partial f_3}{\partial T} \\ \frac{\partial f_4}{\partial C_A} & \frac{\partial f_4}{\partial C_B} & \frac{\partial f_4}{\partial C_C} & \frac{\partial f_4}{\partial C_D} & \frac{\partial f_4}{\partial T} \\ \frac{\partial f_5}{\partial C_A} & \frac{\partial f_5}{\partial C_B} & \frac{\partial f_5}{\partial C_C} & \frac{\partial f_5}{\partial C_D} & \frac{\partial f_5}{\partial T} \end{bmatrix}, \quad J_u = \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial F_f} & \frac{\partial f_1}{\partial T_w} \\ \frac{\partial f_2}{\partial F_f} & \frac{\partial f_2}{\partial T_w} \\ \frac{\partial f_3}{\partial F_f} & \frac{\partial f_3}{\partial T_w} \\ \frac{\partial f_4}{\partial F_f} & \frac{\partial f_4}{\partial T_w} \\ \frac{\partial f_5}{\partial F_f} & \frac{\partial f_5}{\partial T_w} \end{bmatrix}$$

Calculando os valores da matriz jacobiana  $J_x$  para o modelo do reator, tem-se:

$$\frac{\partial f_1}{\partial C_A} = -\frac{1}{\tau} - k_1 e^{(-E_1/T)} - 2k_3 e^{(-E_3/T)} C_A(t)$$

$$\frac{\partial f_2}{\partial C_A} = k_1 e^{(-E_1/T)}$$

$$\frac{\partial f_3}{\partial C_A} = 0$$

$$\frac{\partial f_4}{\partial C_A} = k_3 e^{(-E_3/T)} C_A(t)$$

$$\frac{\partial f_5}{\partial C_A} = -\frac{\Delta H_1 k_1 e^{(-E_1/T)} + \Delta H_3 2k_3 e^{(-E_3/T)} C_A(t)}{\rho c_p}$$

$$\begin{array}{lll}
\frac{\partial f_1}{\partial C_B} = 0 & \frac{\partial f_1}{\partial C_C} = 0 & \frac{\partial f_1}{\partial C_D} = 0 \\
\frac{\partial f_2}{\partial C_B} = \frac{-1}{\tau} - k_2 e^{(-E_2/\tau)} & \frac{\partial f_2}{\partial C_C} = 0 & \frac{\partial f_2}{\partial C_D} = 0 \\
\frac{\partial f_3}{\partial C_B} = k_2 e^{(-E_2/\tau)} & \frac{\partial f_3}{\partial C_C} = -\frac{1}{\tau} & \frac{\partial f_3}{\partial C_D} = 0 \\
\frac{\partial f_4}{\partial C_B} = 0 & \frac{\partial f_4}{\partial C_C} = 0 & \frac{\partial f_4}{\partial C_D} = -\frac{1}{\tau} \\
\frac{\partial f_5}{\partial C_B} = \frac{-\Delta H_2 k_2 e^{(-E_2/\tau)}}{\rho c_p} & \frac{\partial f_5}{\partial C_C} = 0 & \frac{\partial f_5}{\partial C_D} = 0
\end{array}$$

$$\begin{array}{l}
\frac{\partial f_1}{\partial T} = \frac{-(k_1 e^{(-E_1/\tau)} E_1 C_A(t) + k_3 e^{(-E_3/\tau)} E_3 C_A^2(t))}{T^2} \\
\frac{\partial f_2}{\partial T} = \frac{k_1 e^{(-E_1/\tau)} E_1 C_A(t) - k_2 e^{(-E_2/\tau)} E_2 C_B(t)}{T^2} \\
\frac{\partial f_3}{\partial T} = \frac{k_2 e^{(-E_2/\tau)} E_2 C_B(t)}{T^2} \\
\frac{\partial f_4}{\partial T} = \frac{k_3 e^{(-E_3/\tau)} E_3 C_A^2(t)}{2T^2} \\
\frac{\partial f_5}{\partial T} = -\frac{1}{\tau} + \frac{-(\Delta H_1 k_1 e^{(-E_1/\tau)} E_1 C_A(t) + \Delta H_2 k_2 e^{(-E_2/\tau)} E_2 C_B(t) + \Delta H_3 k_3 e^{(-E_3/\tau)} E_3 C_A^2(t))}{T^2 \rho c_p} - \frac{UA_t}{\rho V c_p}
\end{array}$$

Para a matriz  $J_u$ , tem-se:

$$\begin{array}{ll}
\frac{\partial F_1}{\partial F_f} = \frac{C_{Af} - C_A}{V} & \frac{\partial F_1}{\partial T_w} = 0 \\
\frac{\partial F_2}{\partial F_f} = \frac{C_{Bf} - C_B}{V} & \frac{\partial F_2}{\partial T_w} = 0 \\
\frac{\partial F_3}{\partial F_f} = \frac{C_{Cf} - C_C}{V} & \frac{\partial F_3}{\partial T_w} = 0 \\
\frac{\partial F_4}{\partial F_f} = \frac{C_{Df} - C_D}{V} & \frac{\partial F_4}{\partial T_w} = 0 \\
\frac{\partial F_5}{\partial F_f} = \frac{T_f - T}{V} & \frac{\partial F_5}{\partial T_w} = \frac{1}{\rho V c_p}
\end{array}$$

Estas equações são integradas juntamente com as equações de estado por meio do *solver* de ODE no MATLAB, obtendo-se os valores de  $x(t_f)$  e  $W(t_f)$ . Os perfis de estado obtidos e os parâmetros de controle propostos são utilizados para avaliar a função objetivo  $J$  e as restrições  $G(x(t), u(t), t, p)$ .

Para otimização das variáveis de decisão  $\hat{u}$ , foi utilizado o método *fmincon* do MATLAB. Este algoritmo é aplicado no laço externo para atualizar as ações de controle até que ocorra a convergência concluindo assim a otimização.

#### 4.4 Aplicação em Tempo Real

Esta seção tem como objetivo apenas mostrar como é a utilização da estrutura proposta em tempo real descrevendo os passos realizados para a sua implementação.

Toda a estrutura foi desenvolvida em Matlab/Simulink, porém para a aplicação em tempo real, o modelo do processo (a planta), desenvolvido na *s-function* e utilizado para casos *offline*, é substituído pelo mesmo modelo do reator, porém implementado na linguagem do simulador EMSO (Figura 4.4.1). A troca de dados entre os dois *softwares* é feita através do protocolo de comunicação OPC (A implementação do módulo OPC, desenvolvido no Matlab, responsável pela leitura e escrita dos dados no servidor OPC está descrita no Apêndice B).

Na Figura 4.4.1, abaixo, é mostrado como as equações do modelo do reator CSTR com reação de Van der Vusse foram inseridas na linguagem do simulador, demais detalhes da implementação do modelo do reator no EMSO encontram-se no Apêndice C.

```
cstr_vdv_noniso.mso
156 EQUATIONS
157 "Overall Mass Balance"
158   diff(V) = Inlet.Fvol - Outlet.Fvol;
159
160 "Reaction Rate - First Reaction"
161   r1 = k1*Ca;
162
163 "Reaction Rate - Second Reaction"
164   r2 = k2*Cb;
165
166 "Reaction Rate - Third Reaction"
167   r3 = k3*Ca*Ca;
168
169 "A Component Mass Balance"
170   tau*diff(Ca) = -tau*r1 - tau*r3 + Inlet.Ca - Outlet.Ca;
171
172 "B Component Mass Balance"
173   tau*diff(Cb) = tau*r1 - tau*r2 + Inlet.Cb - Outlet.Cb;
174
175 "C Component Mass Balance"
176   tau*diff(Cc) = tau*r2 + Inlet.Cc - Outlet.Cc;
177
178 "D Component Mass Balance"
179   tau*diff(Cd) = tau*r3/2 + Inlet.Cd - Outlet.Cd;
180
181 "Residence Time"
182   tau = V/Inlet.Fvol;
183
184 "Energy Balance"
185   ro * V * cp * diff(T) = Inlet.Fvol * ro * cp * (Inlet.T - T) + qr - q;
186
187 "Heat Transfer Rate"
188   q = U * At * (T - Tw);
189
190 "Reaction Heat Rate"
191   qr = -V * (H1*r1 + H2*r2 + H3*r3);
192
193 "Reactor Volume"
194   V = A * h;
195
196 "Heat Transfer Area"
197   At = A + pi*D*h;
198
199 "Valve Equation"
200   Outlet.Fvol = Cv * sqrt(h);
201
202 end
```

Figura 4.4.1 – Equações do modelo do reator CSTR com cinética de Van der Vusse na linguagem do simulador de processos EMSO.

### Comunicação dos Dados via OPC

A vantagem de se utilizar o simulador EMSO é que ele possui uma interface EMSO-OPC, sendo possível fazer a configuração “de – para” entre os sinais das variáveis de processo gerados por este simulador com o servidor OPC funcionando assim como a planta do processo em tempo real. Na Figura 4.4.2 , é apresentado como esta configuração foi feita no presente estudo de caso.

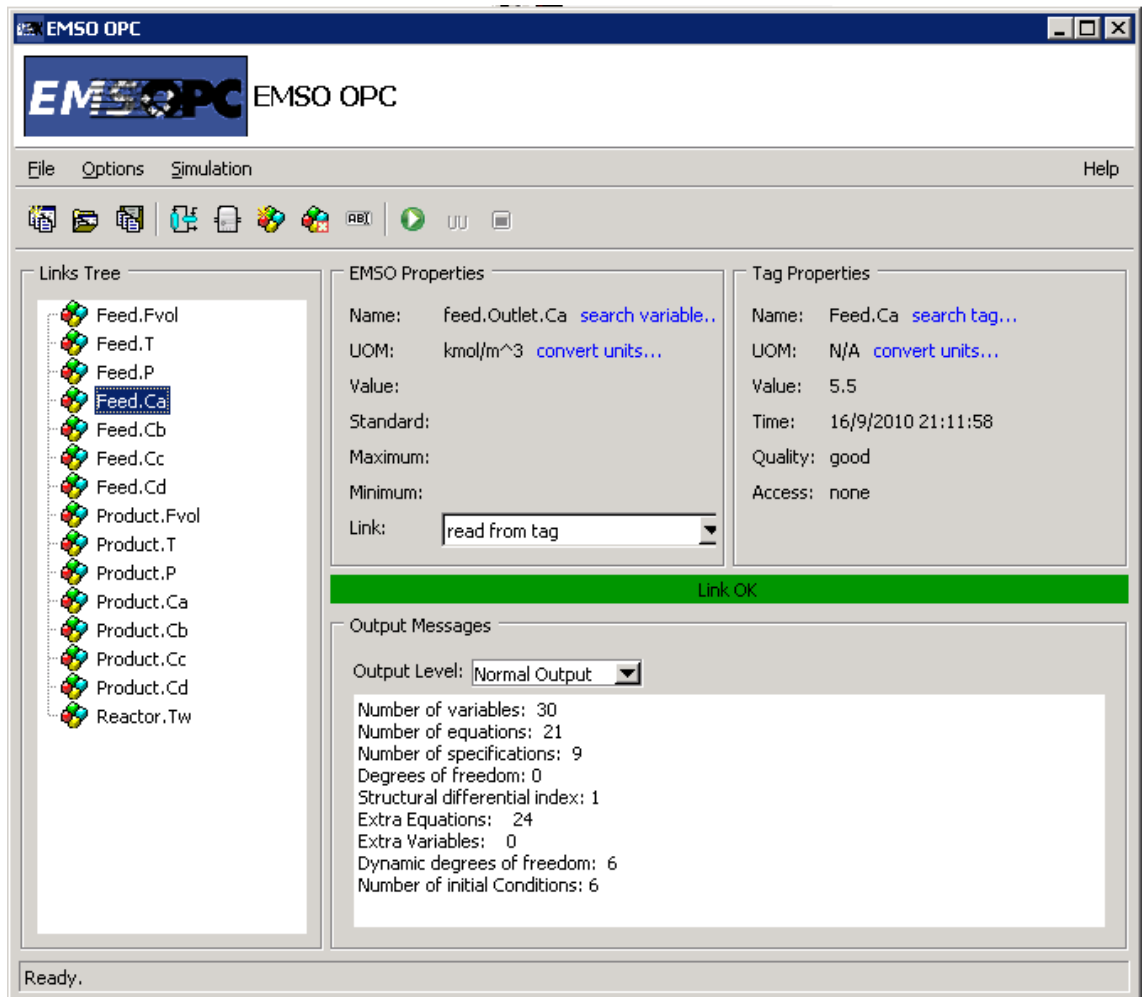


Figura 4.4.2 – Configuração da interface EMSO-OPC.

Já no lado do servidor OPC, para finalizar a comunicação, foram criados os *tags* para armazenar e fornecer os dados da planta para o sistema de otimização dinâmica no Matlab/Simulink (Figura 4.4.3).

Time	Variable	Value
22:00:00	Feed.Fvol	10.00
18:11:58	Feed.Ca	5.50
22:00:00	Feed.Cb	0.00
22:00:00	Feed.Cc	0.00
22:00:00	Feed.Cd	0.00
22:00:00	Feed.T	450.00
22:00:00	Reactor.Tw	700.00
22:00:00	Feed.P	1.00
22:00:00	Product.Fvol	0.00
22:00:00	Product.T	0.00
22:00:00	Product.P	0.00
22:00:00	Product.Ca	0.00
22:00:00	Product.Cb	0.00
22:00:00	Product.Cc	0.00
22:00:00	Product.Cd	0.00

Figura 4.4.3 – Configuração do Servidor OPC para estudo de casos do reator CSTR com reação de Van der Vusse.

O protocolo OPC é muito utilizado pelas indústrias de processos para conectar ferramentas computacionais com os dispositivos do sistema de controle. A configuração apresentada acima mostra como é possível a utilização da estrutura proposta na aplicação em tempo real a uma planta de processo.



## 5 Resultados e Discussões

As variáveis de controle do sistema são a vazão de entrada  $F_f$  e a temperatura da corrente de resfriamento  $T_w$ . A variável de otimização é a concentração do produto no tempo final do horizonte da otimização,  $C_B(t_f)$ . As oportunidades de otimização neste estudo estão ligadas às perturbações na concentração do reagente  $C_{Af}$  e na temperatura da corrente de entrada  $T_f$ . Com objetivo de avaliar o desempenho do sistema frente a estas perturbações, dois dos principais módulos da estrutura proposta foram avaliados:

### 5.1 Avaliação do Módulo de Otimização Dinâmica

### 5.2 Avaliação do Módulo Disparador D-RTO

## 5.1 Avaliação do Módulo de Otimização Dinâmica

Este estudo teve como principal objetivo avaliar a influência do módulo da otimização dinâmica no desempenho do sistema de D-RTO proposto. Para efeito de comparação, realizou-se um planejamento de perturbações e testou-se a estrutura primeiramente com o módulo da otimização dinâmica ativado e em seguida desativado, ou seja, somente com o módulo MPC operando no sistema de controle. É importante destacar que, em ambas as situações, partiu-se das condições ótimas e, os distúrbios não são medidos para o MPC. Os parâmetros utilizados na configuração do controlador para ambos os casos foram:

- . Tempo de amostragem ( $t.a.$ ): 1.5 min
- . Horizonte de Predição: 20  $t.a.$

- . Horizonte de Controle: 5 *t.a.*
- . Matriz de pesos nas variáveis manipuladas: [0.1, 0.1]
- . Matriz de pesos na taxa de atualização das variáveis manipuladas: [1, 1]
- . Matriz de pesos nas variáveis controladas: [40, 40]

Os estudos de casos de planejamento de perturbações encontram-se resumidos na Tabela 5.1.1 abaixo:

Tabela 5.1.1 – Planejamento de perturbações para avaliação da camada de otimização

<b>Casos</b>	<b><i>t</i> &lt; 50min</b>	<b><i>t</i> ≥ 50min</b>
1. Sem perturbação	$C_{Af} = 5,5\text{kmol/l}$ $T_f = 378,05\text{K}$	$C_{Af} = 5,5\text{kmol/l}$ $T_f = 378,05\text{K}$
2. Perturbação na concentração do reagente	$C_{Af} = 5,5\text{kmol/l}$	$C_{Af} = 3,5\text{kmol/l}$
3. Perturbação na concentração do reagente	$C_{Af} = 5,5\text{kmol/l}$	$C_{Af} = 7,5\text{kmol/l}$
4. Perturbação na temperatura de entrada	$T_f = 378,05\text{K}$	$T_f = 368,05\text{K}$
5. Perturbação na temperatura de entrada	$T_f = 378,05\text{K}$	$T_f = 388,05\text{K}$

Os resultados das simulações foram comparados e encontram-se abaixo com os comentários da análise para cada caso.

### **Caso 1: Sem Perturbação**

Neste caso, não é realizada perturbação alguma e, portanto, o *setpoint* final alcançado pelo controlador é o mesmo para ambos os casos. Note, porém que o otimizador traça um perfil da trajetória de referência para o componente *B* (Figura 5.1.1) e também para a temperatura *T* (Figura 5.1.2) partindo do instante inicial até alcançar seu valor máximo de forma exponencial, ao passo que no caso sem o

otimizador a simulação é realizada partindo com o valor do *setpoint* já definido em seu máximo. Esta diferença demonstra um aumento significativo no desempenho do MPC ao receber uma trajetória de referência obtida pelo D-RTO, que pode ser comprovado com um menor *overshooting* e menor tempo de assentamento, em torno de 17min para  $C_B$  e para  $T$ , enquanto que para o caso sem o otimizador, este tempo chega a 40min para  $C_B$  e aproximadamente 50min para  $T$ . Nas Figuras 5.1.3 a 5.1.5 são apresentadas as demais concentrações, onde também observa-se o efeito do maior tempo de assentamento para o caso sem D-RTO. As ações de controle, apresentadas nas Figuras 5.1.6 e 5.1.7, mostram que o melhor desempenho do D-RTO não foi à custa de ações mais agressivas dos controladores, mas sim da trajetória de controle proposta pela otimização dinâmica.

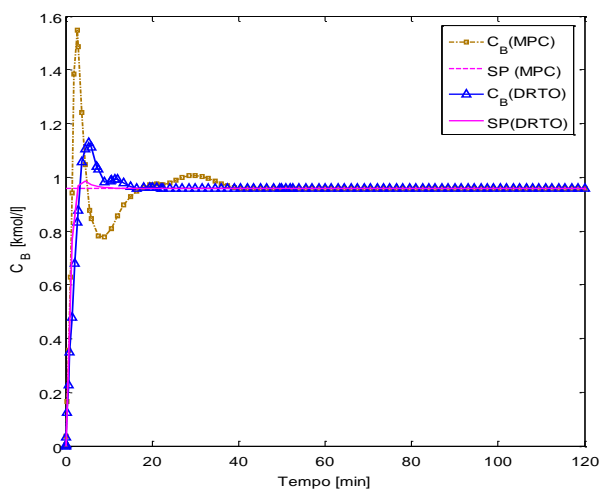


Figura 5.1.1- Concentração de B sem perturbação com D-RTO desativado (MPC) e ativado.

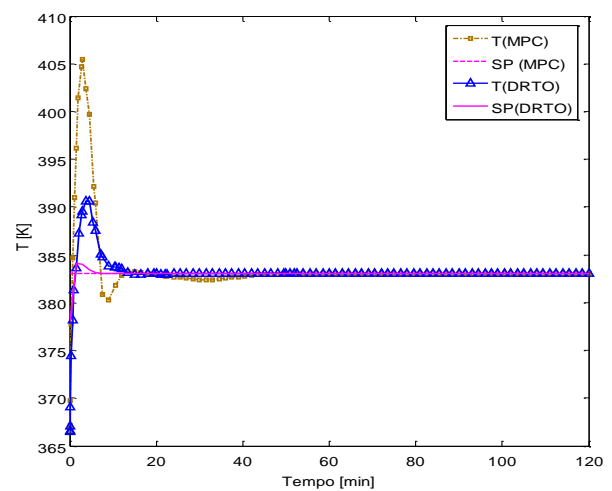


Figura 5.1.2 – Temperatura do reator sem perturbação com D-RTO desativado (MPC) e ativado.

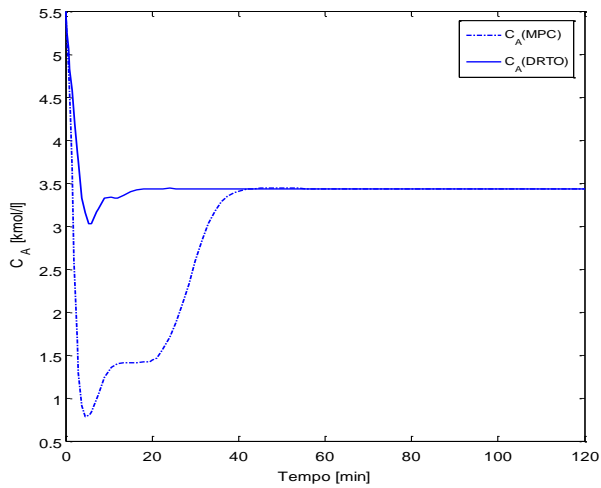


Figura 5.1.3- Concentração de A sem perturbação com D-RTO desativado (MPC) e ativado.

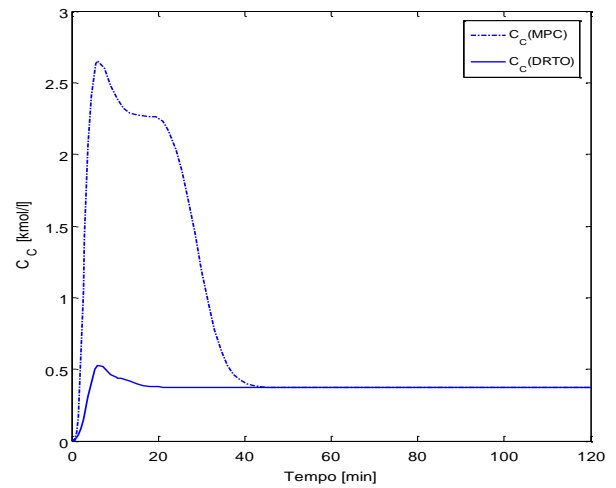


Figura 5.1.4- Concentração de C sem perturbação com D-RTO desativado (MPC) e ativado.

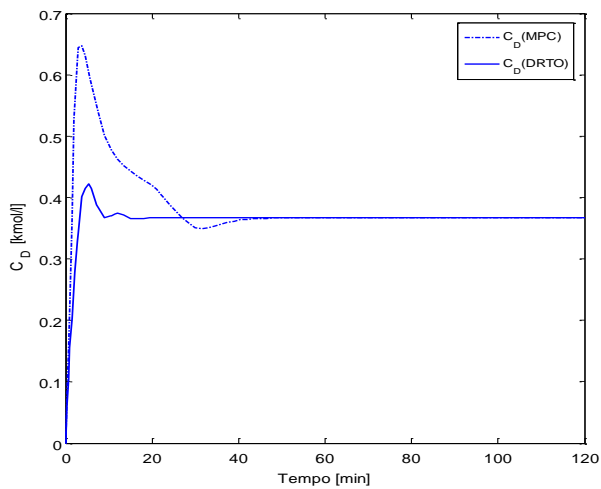


Figura 5.1.5- Concentração de D sem perturbação com D-RTO desativado (MPC) e ativado.

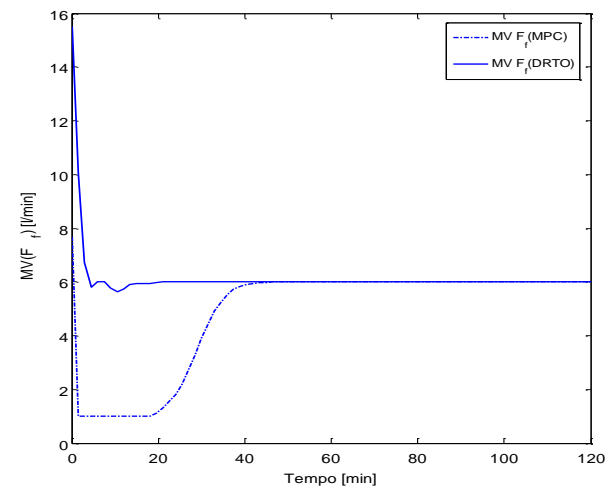


Figura 5.1.6- Vazão de alimentação sem perturbação com D-RTO desativado (MPC) e ativado.

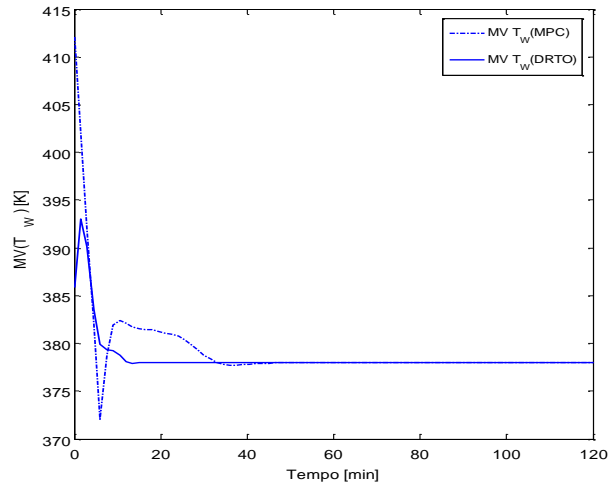


Figura 5.1.7- Temperatura de resfriamento sem perturbação com D-RTO desativado (MPC) e ativado.

**Caso 2: Perturbação na concentração do reagente:  $C_{Af}(t_2) = C_{Af}(t_1) - 2\text{kmol/l}$**

No Caso 2, ocorre a perturbação negativa na concentração do reagente A na corrente de alimentação. Pode-se observar que no momento da perturbação o otimizador imediatamente traça uma nova trajetória de referência para o MPC, ao passo que sem o D-RTO o controlador tenta retornar, sem sucesso, à concentração de B para o *setpoint* original (Figura 5.1.8), deixando um *offset*, que é resultado da restrição inferior imposta na vazão de alimentação (Figura 5.1.13). Outro ponto importante neste caso é o aumento excessivo na produção dos subprodutos C e D para o caso sem o D-RTO (Figura 5.1.11 e Figura 5.1.12).

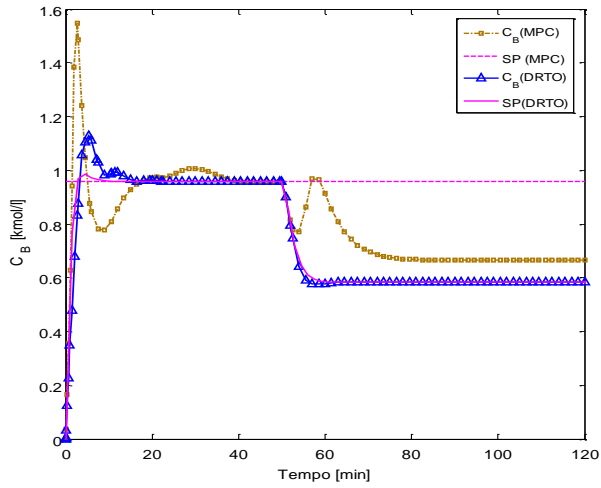


Figura 5.1.8- Concentração de B, perturbação caso 2, com D-RTO desativado (MPC) e ativado.

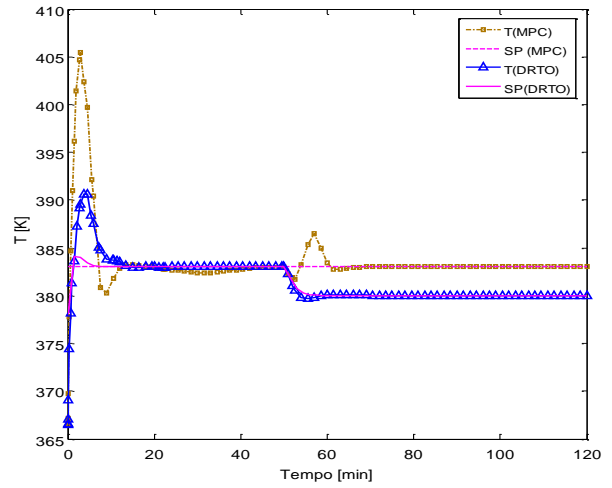


Figura 5.1.9 – Temperatura do reator, perturbação caso 2, com D-RTO desativado (MPC) e ativado.

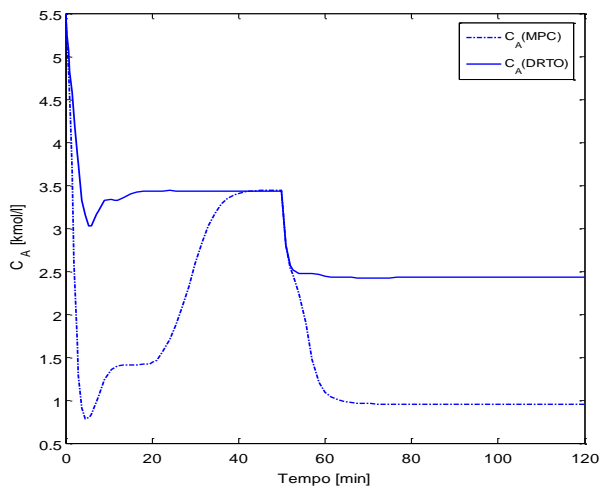


Figura 5.1.10- Concentração de A, perturbação caso 2, com D-RTO desativado (MPC) e ativado.

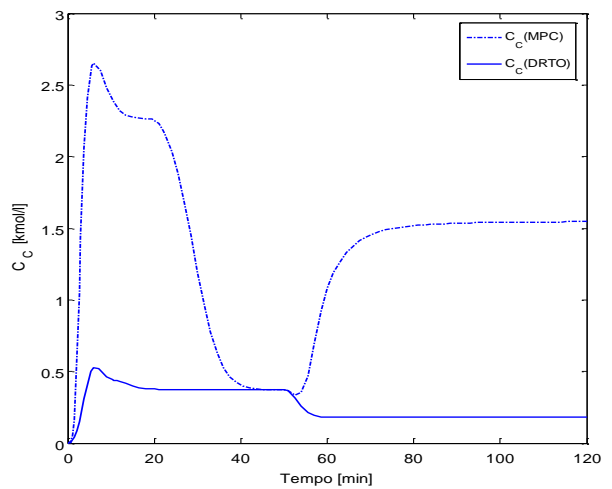


Figura 5.1.11- Concentração de C, perturbação caso 2, com D-RTO desativado (MPC) e ativado.

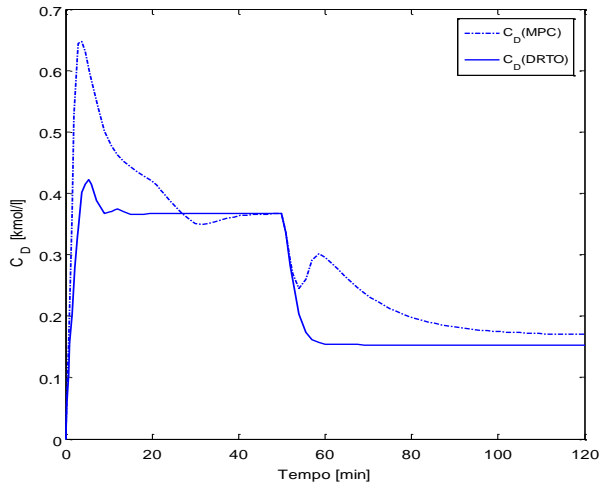


Figura 5.1.12- Concentração de D, perturbação caso 2, com D-RTO desativado (MPC) e ativado.

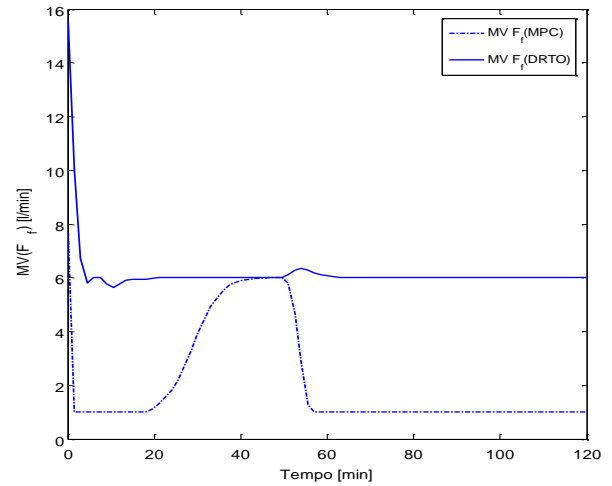


Figura 5.1.13- Vazão de alimentação, perturbação caso 2, com D-RTO desativado (MPC) e ativado.

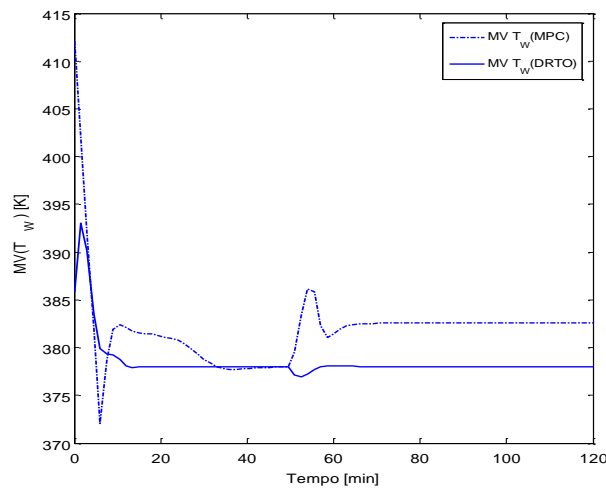


Figura 5.1.14- Temperatura de resfriamento, perturbação caso 2, com D-RTO desativado (MPC) e ativado.

**Caso 3: Perturbação na concentração do reagente:  $C_{A\beta}(t_2) = C_{A\beta}(t_1) + 2\text{kmol/l}$**

Neste caso há um aumento de  $2\text{kmol/l}$  na concentração do componente A. Nas Figuras 5.1.15 e 5.1.16, observa-se que o D-RTO, no momento da perturbação, recalcula a nova trajetória de referência visando maximizar a concentração de B

chegando a um valor no tempo final superior. Esta mesma oportunidade não é aproveitada na situação sem o D-RTO, uma vez que o *setpoint* está fixo, o MPC retorna as variáveis controladas aos seus valores originais.

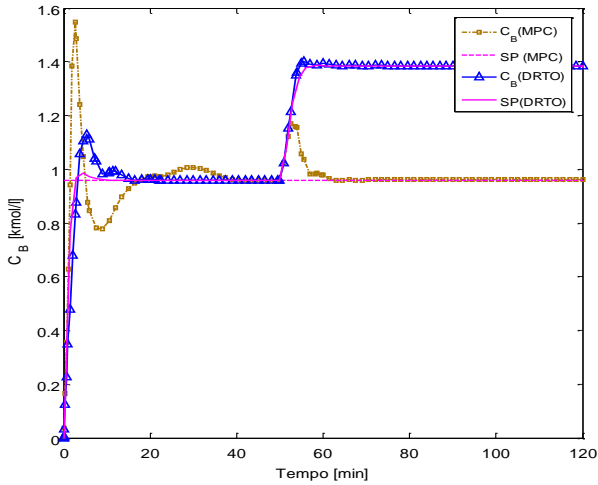


Figura 5.1.15- Concentração de B, perturbação caso 3, com D-RTO desativado (MPC) e ativado.

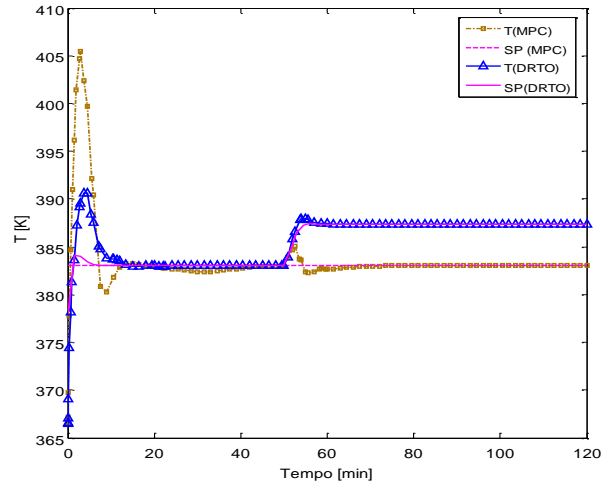


Figura 5.1.16- Temperatura do reator, perturbação caso 3, com D-RTO desativado (MPC) e ativado.

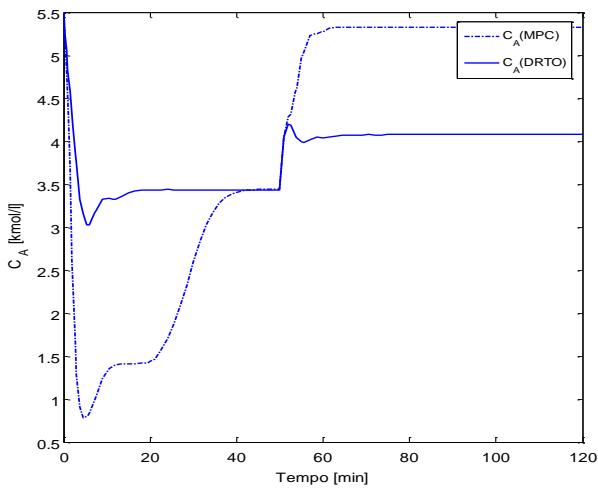


Figura 5.1.17- Concentração de A, perturbação caso 3, com D-RTO desativado (MPC) e ativado.

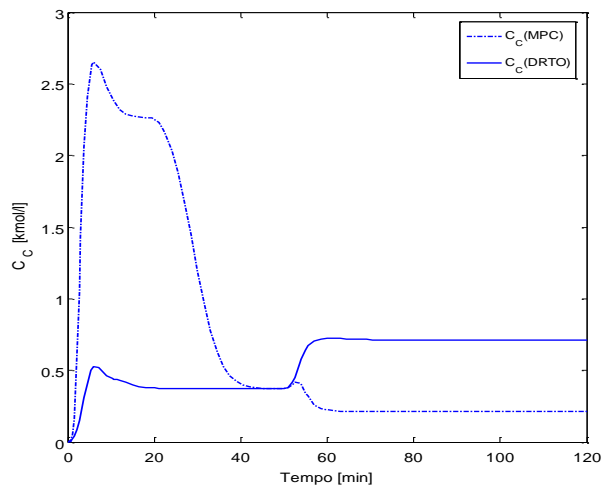


Figura 5.1.18- Concentração de C, perturbação caso 3, com D-RTO desativado (MPC) e ativado.



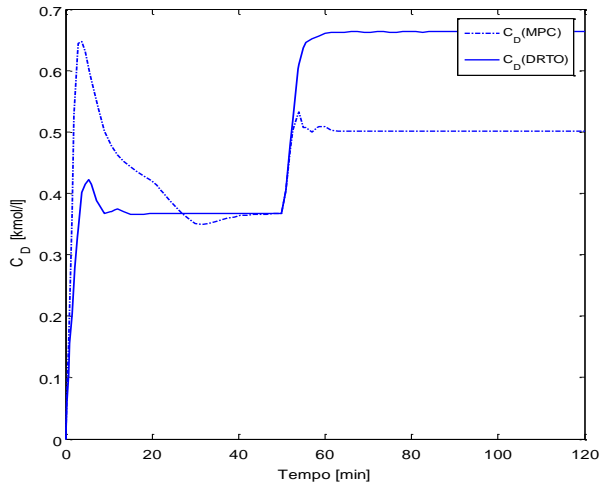


Figura 5.1.19- Concentração de D, perturbação caso 3, com D-RTO desativado (MPC) e ativado.

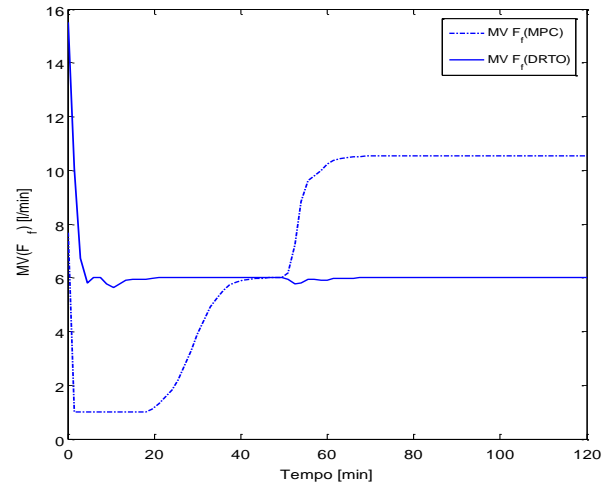


Figura 5.1.20- Vazão de alimentação, perturbação caso 3, com D-RTO desativado (MPC) e ativado.

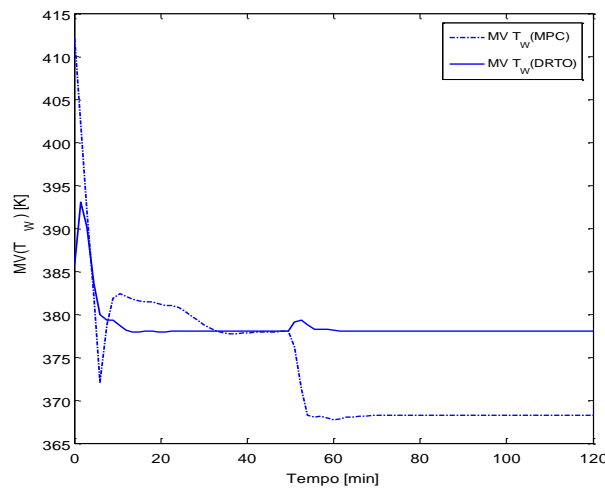


Figura 5.1.21- Temperatura de resfriamento, perturbação caso 3, com D-RTO desativado (MPC) e ativado.

Nas Figuras 5.1.20 e 5.1.21, percebe-se, no caso do MPC, que as variáveis manipuladas atuaram de forma a rejeitar a perturbação na concentração de entrada, ficando a temperatura em um nível mais baixo e a vazão de alimentação em um nível mais alto após a perturbação para compensar o aumento da concentração da alimentação (ambas no sentido de reduzir a taxa de geração do produto B). No caso do D-RTO, o

aumento da concentração da alimentação é aproveitado para maximizar a produção do produto desejado, sem a necessidade de mover o valor final das variáveis manipuladas de seu ponto, mas apenas no transiente inicial após a perturbação de forma a reduzir o tempo da transição para o próximo *setpoint* de  $C_B$ .

**Caso 4: Perturbação na temperatura:  $T_f(t_2) = T_f(t_1) - 10K$**

Neste caso, há um decréscimo na temperatura em  $10K$ , o que naturalmente provoca também uma diminuição na concentração do componente  $B$ . Avaliando as Figuras 5.1.22 a 5.1.28, uma observação interessante é que houve uma boa rejeição ao distúrbio por parte do D-RTO, que manteve o valor do *setpoint* original tanto para  $C_B$  quanto para  $T$ , sendo a condição ótima restabelecida apesar da perturbação.

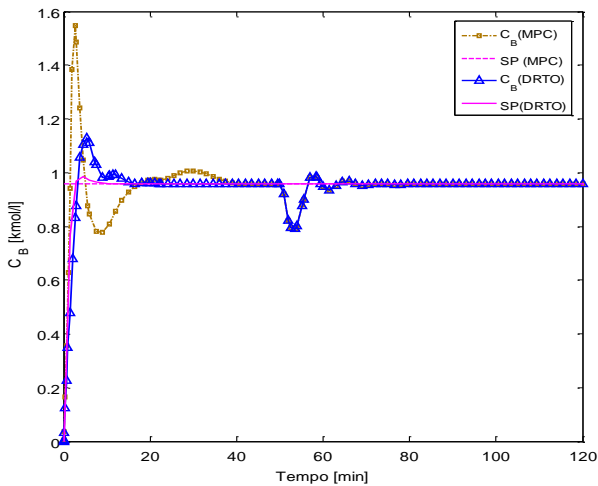


Figura 5.1.22- Concentração de B, perturbação caso 4, com D-RTO desativado (MPC) e ativado.

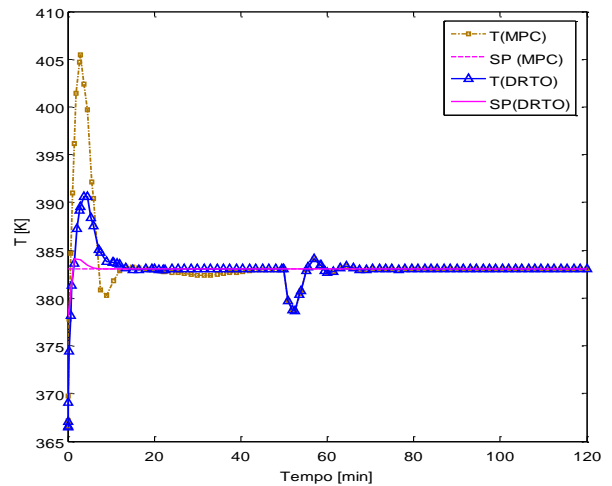


Figura 5.1.23- Temperatura do reator, perturbação caso 4, com D-RTO desativado (MPC) e ativado.

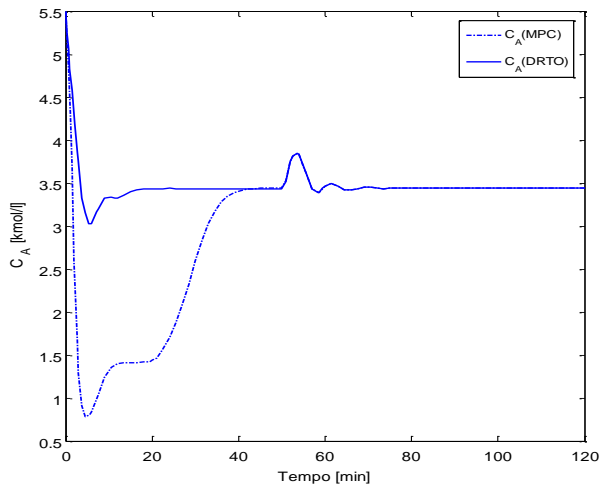


Figura 5.1.24- Concentração de A, perturbação caso 4, com D-RTO desativado (MPC) e ativado.

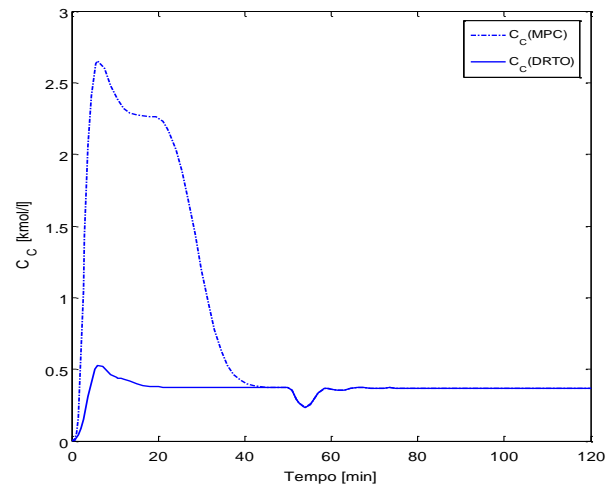


Figura 5.1.25- Concentração de C, perturbação caso 4, com D-RTO desativado (MPC) e ativado.

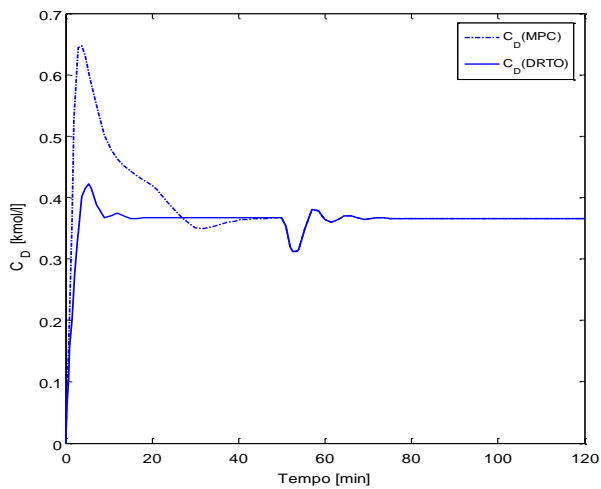


Figura 5.1.26- Concentração de D, perturbação caso 4, com D-RTO desativado (MPC) e ativado.

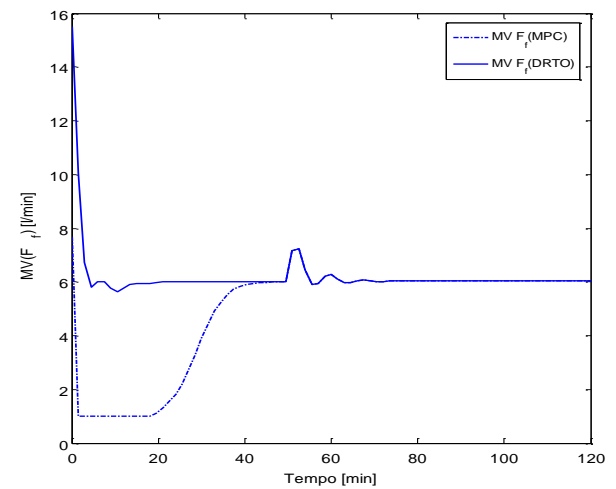


Figura 5.1.27- Vazão de alimentação, perturbação caso 4, com D-RTO desativado (MPC) e ativado.

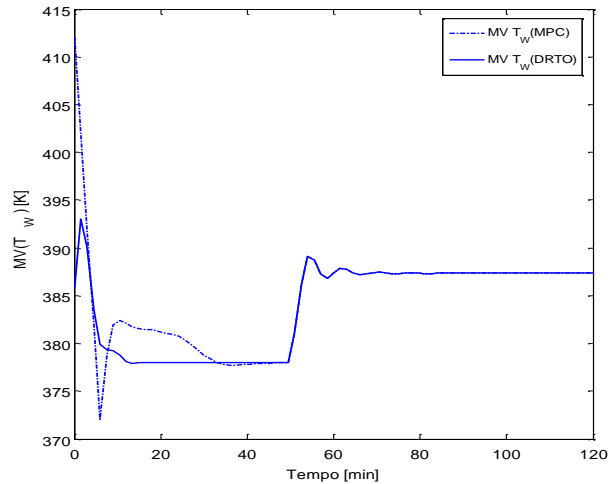


Figura 5.1.28- Temperatura de resfriamento, perturbação caso 4, com D-RTO desativado (MPC) e ativado.

**Caso 5: Perturbação na temperatura:  $T_f(t_2) = T_f(t_1) + 10K$**

No Caso 5, ocorre um aumento na temperatura na corrente de alimentação. De forma análoga ao Caso 3, as mesmas observações podem ser feitas para as Figuras 5.1.29 e 5.1.30, ou seja, através desta oportunidade o otimizador consegue recalcular um novo *setpoint* tanto para o componente *B* quanto para a temperatura do reator *T* acima do valor original. Repare, na Figura 5.1.35, que o controlador conseguiu atingir este novo objetivo mantendo a temperatura da corrente de resfriamento praticamente constante, ao passo que, no caso sem o D-RTO, o MPC diminui a temperatura de resfriamento para trazer a temperatura do reator ao seu valor original, o que, em termos práticos, tendo em vista a função objetivo, representaria um custo adicional desnecessário para a indústria.

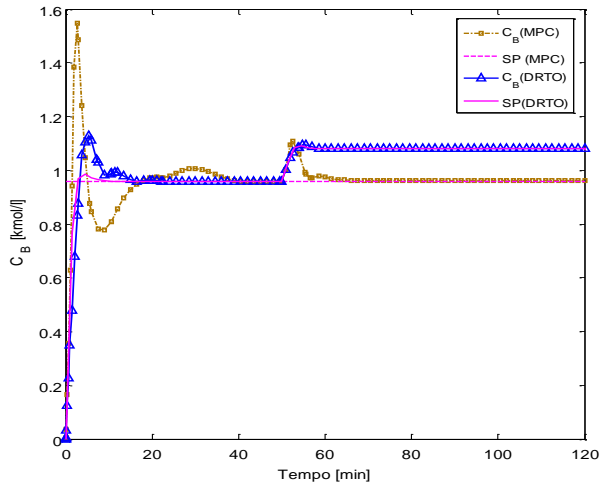


Figura 5.1.29- Concentração de B, perturbação caso 5, com D-RTO desativado (MPC) e ativado.

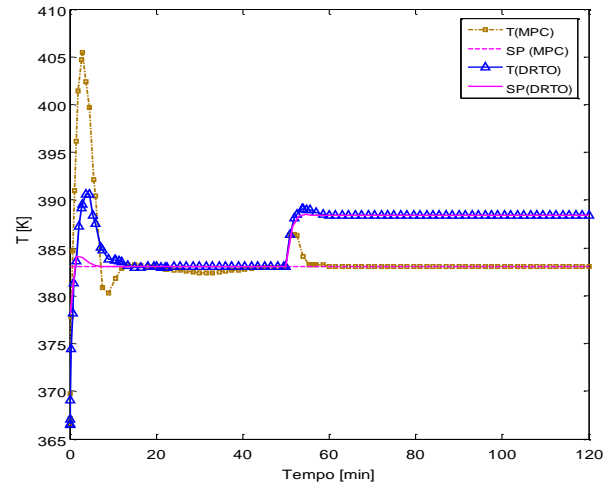


Figura 5.1.30- Temperatura do reato, perturbação caso 5, com D-RTO desativado (MPC) e ativado.

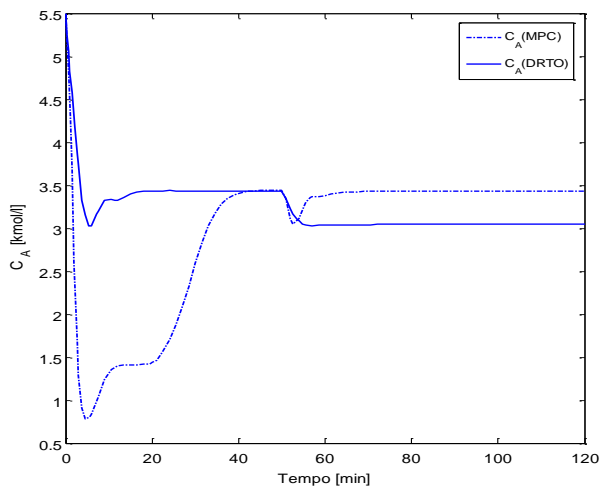


Figura 5.1.31- Concentração de A, perturbação caso 5, com D-RTO desativado (MPC) e ativado.

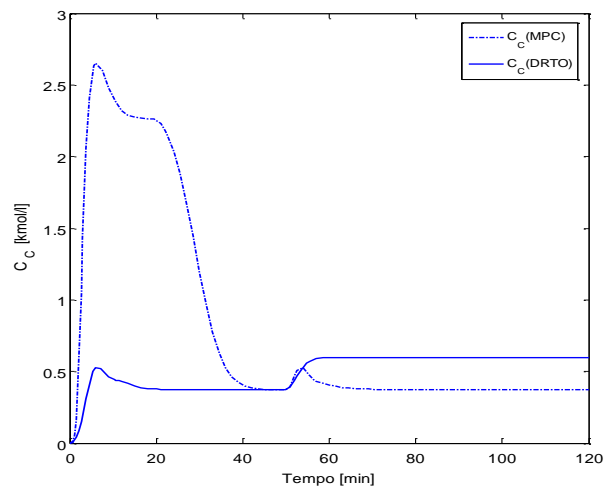


Figura 5.1.32- Concentração de C, perturbação caso 5, com D-RTO desativado (MPC) e ativado.

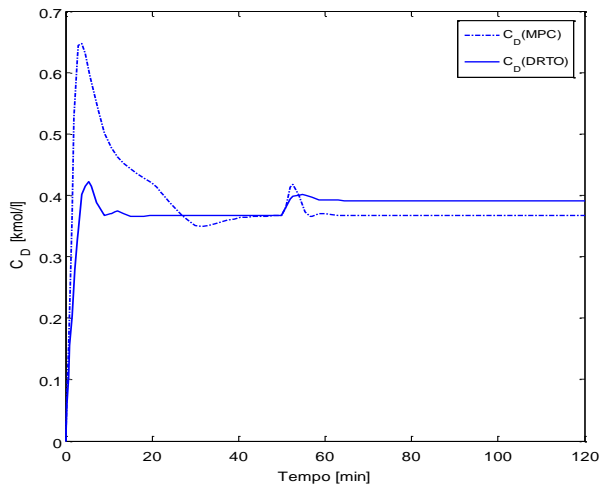


Figura 5.1.33- Concentração de D, perturbação caso 5, com D-RTO desativado (MPC) e ativado.

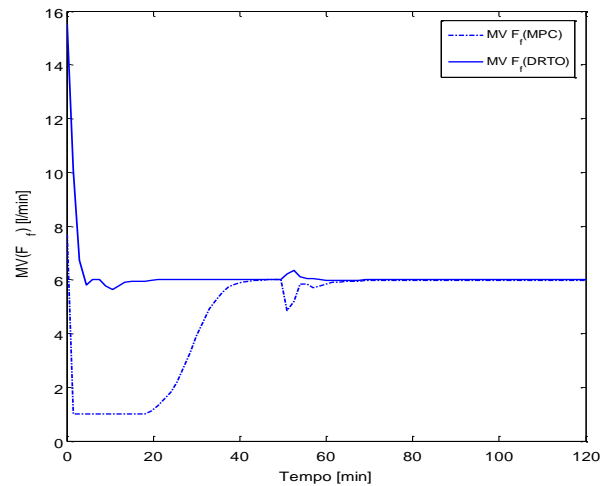


Figura 5.1.34- Vazão de alimentação, perturbação caso 5, com D-RTO desativado (MPC) e ativado.

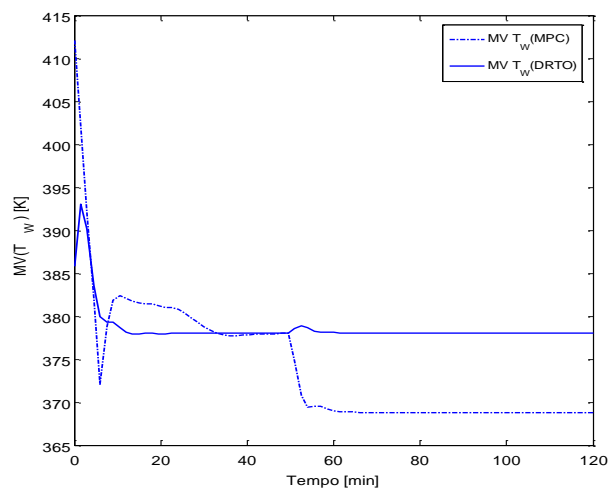


Figura 5.1.35- Temperatura de resfriamento, perturbação caso 5, com D-RTO desativado (MPC) e ativado.

## 5.2 Avaliação do Módulo Disparador D-RTO

O desempenho de um sistema de otimização dinâmica em tempo real está diretamente relacionado ao custo computacional gasto durante a otimização.

Uma das formas mais simples de melhorar este desempenho é introduzindo um módulo capaz de perceber os melhores momentos para realizar a otimização, desta forma evita-se a execução desnecessária do D-RTO.

A presença do módulo do disparador D-RTO na estrutura foi estudada realizando uma comparação entre duas formas de programação de execução do D-RTO, a primeira executando apenas de forma cíclica e, a segunda, utilizando o Disparador D-RTO.

O estudo de caso para avaliação do Disparador D-RTO foi realizado frente às seguintes condições:

1. Durante a partida, a concentração de entrada do reagente  $C_{Af}$  aumenta de forma linear de zero a  $5,5\text{kmol/l}$  e, em seguida, permanece constante até o final, conforme mostrado na Figura 5.2.1.

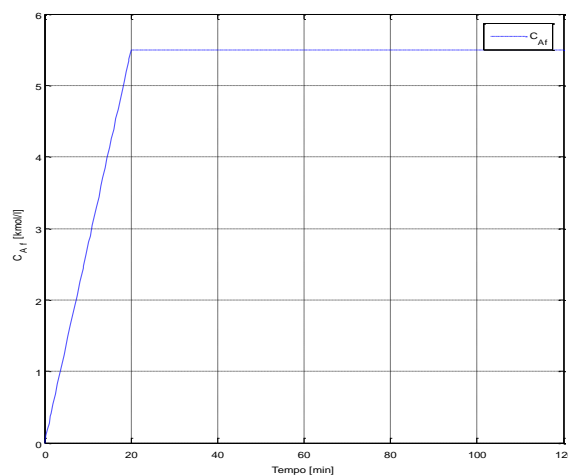


Figura 5.2.1- Perfil da concentração de A na corrente de alimentação.

2. Perturbação na temperatura da corrente de alimentação:  $T_f(t=60min) = T_f(t=0) + 20K$

A execução apenas de forma cíclica foi realizada a cada 17 tempos de amostragem (*t.a.*), já no Disparador D-RTO foram configurados dois modos de execução: o primeiro durante a partida (modo *StartUp*) e o outro após a partida (modo normal) conforme descritos na Tabela 5.2.1.

Tabela 5.2.1 – Estudo de casos com Disparador de DRTO

<b>Casos</b>	<b>Programação</b>
1. Forma cíclica	A cada 17 <i>t.a.</i>
2. Disparador D-RTO	. Modo <i>StartUp</i> : a cada 4.5 min. . Modo normal (após a partida): a cada 17 <i>t.a.</i> e somente quando a variável controlada estiver fora do (set-point + $\Delta$ ) ou quando ocorrer perturbação.

Os parâmetros de sintonia do controlador foram os mesmos utilizados no estudo de caso anterior (seção 5.1).

## Resultados

A Tabela 5.2.2 mostra a quantidade de execuções  $Q_E$  do módulo D-RTO, o tempo total da simulação  $t_{total}$  e o log gerado pelo módulo D-RTO em cada simulação.

Tabela 5.2.2 – Resultados da avaliação do módulo Disparador de DRTO

	$Q_E$	$t_{total}$	<i>Log do módulo D-RTO</i>
1. Forma Cíclica	4	33s	'Opt called on 25.50000 - Reason: Cycle Criteria' 'Opt called on 51.00000 - Reason: Cycle Criteria' 'Opt called on 76.50000 - Reason: Cycle Criteria' 'Opt called on 102.00000 - Reason: Cycle Criteria'
2. Disparador D-RTO	5	31s	'Opt called on 4.50000 - Reason: Start-Up' 'Opt called on 9.00000 - Reason: Start-Up' 'Opt called on 13.50000 - Reason: Start-Up' 'Opt called on 18.00000 - Reason: Start-Up' 'Opt called on 60.00000 - Reason: MD Criteria'



### Caso 1: Forma Cíclica

Os resultados rodando de forma cíclica encontram-se resumidos nas Figuras 5.2.2 a 5.2.8. Observa-se que ainda é possível atenuar mais as ações de controle adicionando pesos nas taxas de variação das variáveis manipuladas, porém esta ação não interfere no resultado/objetivo desta avaliação. O D-RTO foi executado nos tempo 25min, 51min, 76.5min e 102min conforme programação descrita na Tabela 5.2.1.

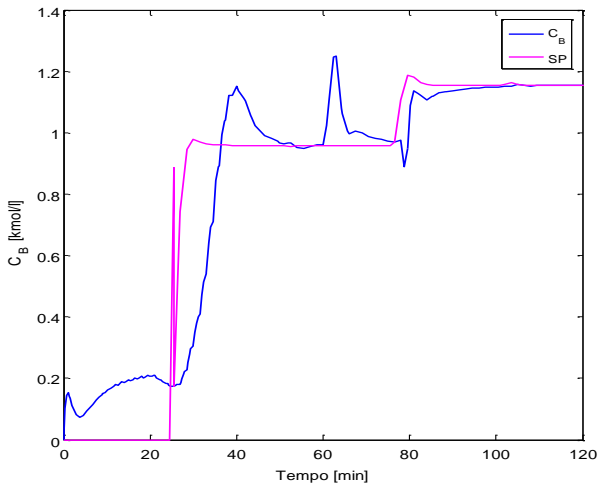


Figura 5.2.2- Concentração de B caso 1: forma cíclica.

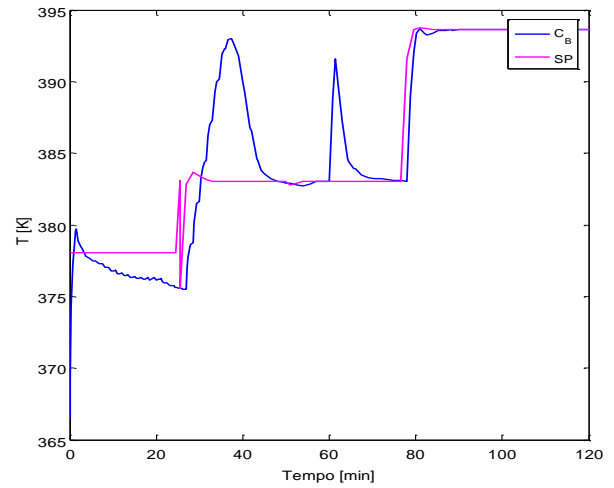


Figura 5.2.3 – Temperatura do reator caso 1: forma cíclica.

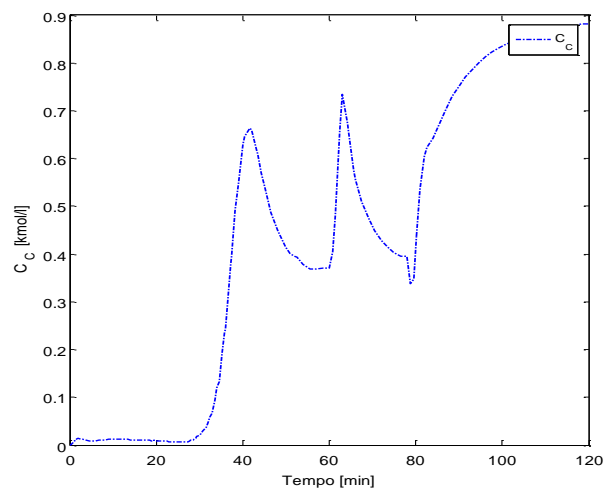
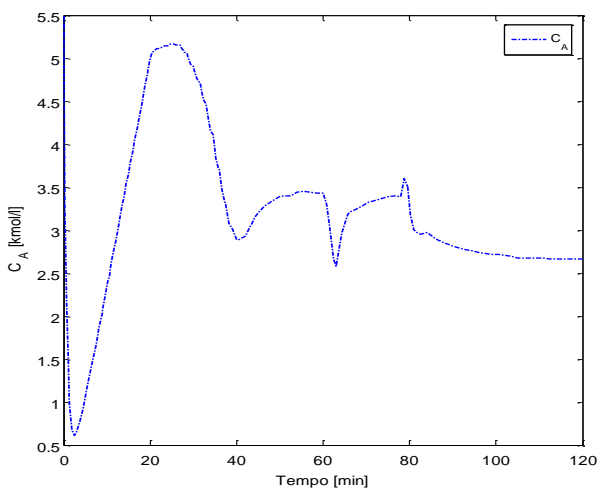


Figura 5.2.4- Concentração de A caso 1: forma cíclica.

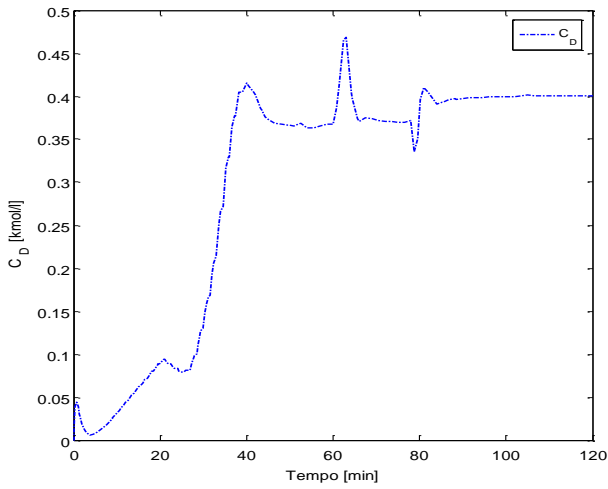


Figura 5.2.5- Concentração de C caso 1: forma cíclica.

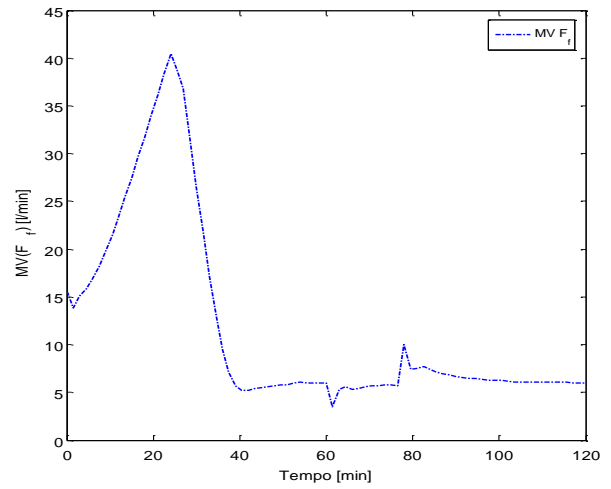


Figura 5.2.6- Concentração de D caso 1: forma cíclica.

Figura 5.2.7- Vazão de alimentação caso 1: forma cíclica.

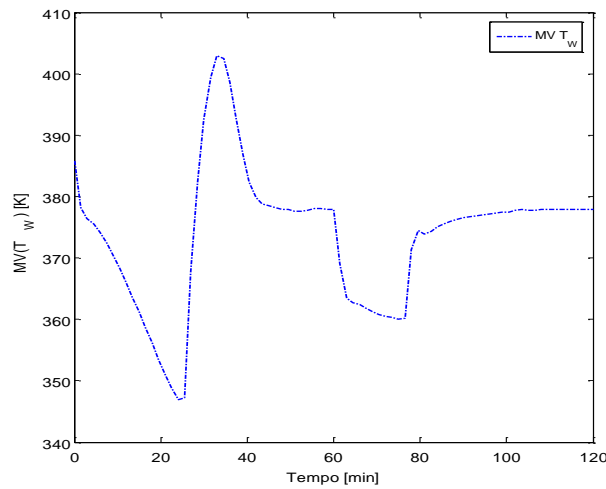


Figura 5.2.8- Temperatura de resfriamento caso 1: forma cíclica.

### **Caso 2: Disparador D-RTO**

Os resultados utilizando o Disparador D-RTO encontram-se resumidos nas Figuras 5.2.9 a 5.2.15. Seguindo a programação descrita na Tabela 5.2.1, o D-RTO foi

executado nos tempo  $4.5min$ ,  $9min$ ,  $13.5min$ ,  $18min$  durante o *startup* e no tempo  $60min$  devido à perturbação.

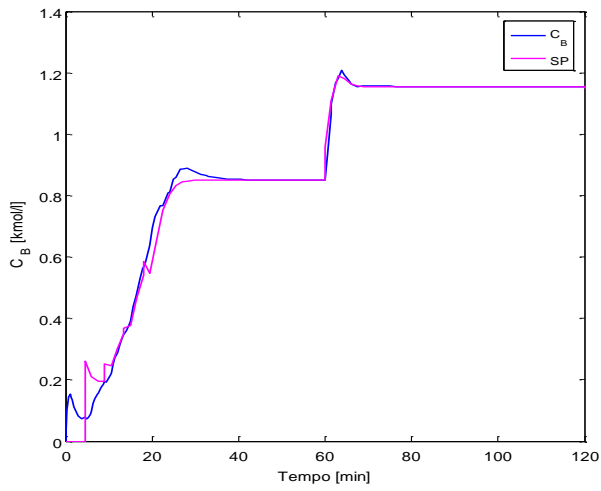


Figura 5.2.9- Concentração de B caso 2: forma inteligente.

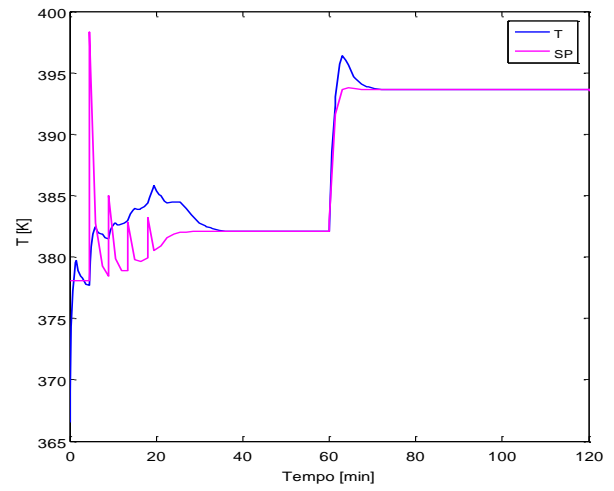


Figura 5.2.10 – Temperatura do reator caso 2: forma inteligente.

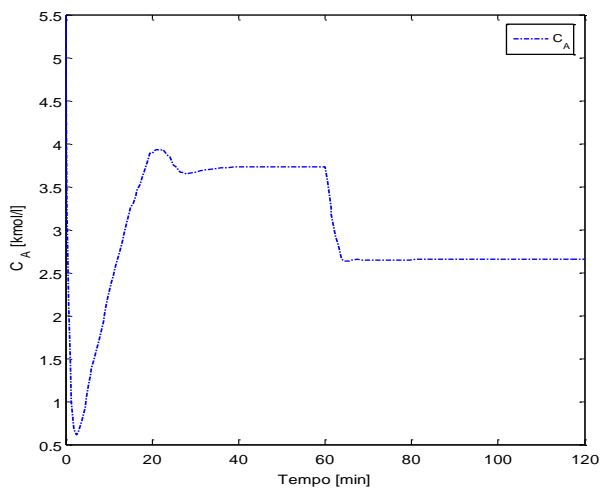


Figura 5.2.11- Concentração de A caso 2: forma inteligente.

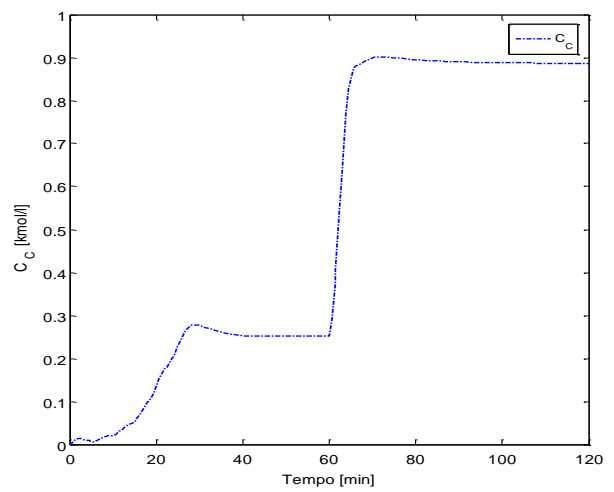


Figura 5.2.12- Concentração de C caso 2: forma inteligente.

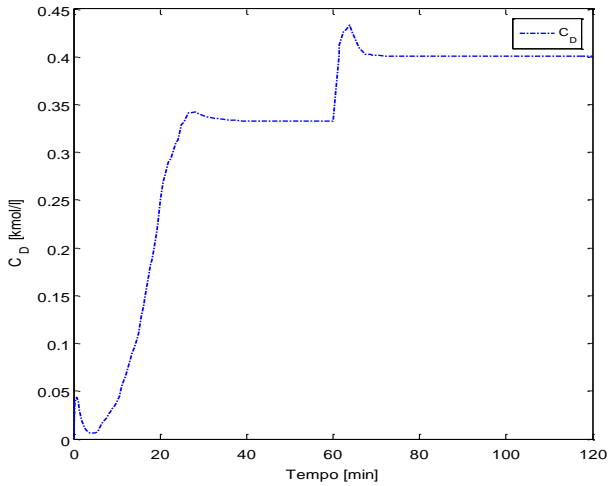


Figura 5.2.13- Concentração de D caso 2: forma inteligente.

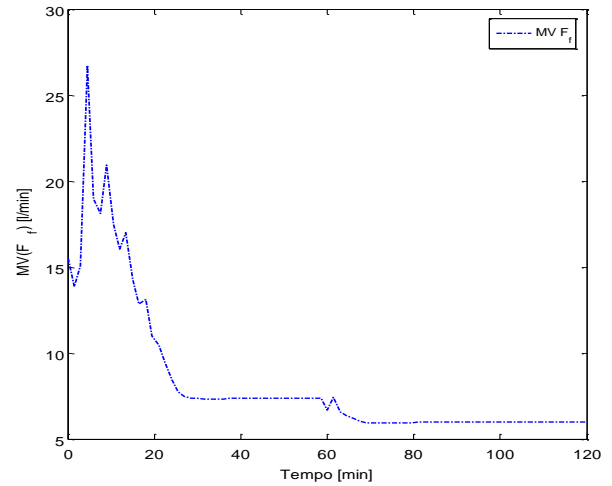


Figura 5.2.14- Vazão de alimentação caso 2: forma inteligente.

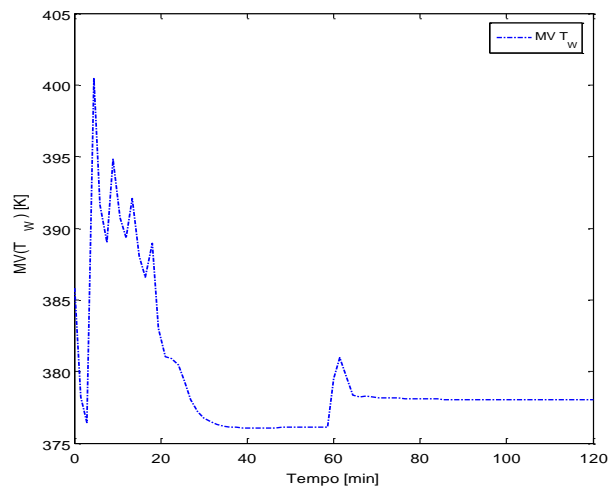


Figura 5.2.15- Temperatura de resfriamento caso 2: forma inteligente.

## Análise dos Resultados

Analisando os resultados obtidos em ambas as programações de execução do D-RTO, pode ser observado que o desempenho do sistema utilizando a forma de execução de maneira inteligente, configurada no módulo Disparador D-RTO, foi muito superior à programação de execução apenas de forma cíclica.

Durante a partida o Disparador D-RTO com inteligência manteve uma frequência de execuções maior gerando um perfil da trajetória de referência muito mais resolvido em comparação ao gerado pela forma cíclica (a cada 17 *t.a.*) que teve a otimização iniciada apenas no tempo 25min, única vez durante o *startup*, causando um maior *overshooting* e, conseqüentemente, maior tempo de assentamento.

Após a partida, o Disparador D-RTO passou a trabalhar no seu modo normal, sendo executado apenas quando ocorreu a perturbação na temperatura da alimentação no tempo 60 *min*. A nova trajetória de referência foi calculada imediatamente após a perturbação o que mostra uma grande vantagem na utilização do Disparador D-RTO visto que a otimização sendo executada apenas da forma cíclica somente percebe a oportunidade de otimizar aproximadamente 20 minutos após o distúrbio, representando um custo econômico adicional, tendo em vista a função objetivo que é maximizar a concentração do produto *B*.

## 6 Conclusões

Com o objetivo de contribuir com a consolidação da tecnologia de D-RTO, neste trabalho foi proposta uma estrutura de otimização dinâmica em tempo real para operar nas plantas de processo. Esta estrutura foi testada em um reator CSTR com reação de Van der Vusse e as oportunidades de otimização frente a perturbações no processo foram exploradas avaliando dois módulos da estrutura: o módulo da otimização dinâmica e o módulo Disparador D-RTO.

Um novo conceito de disparador de D-RTO foi apresentado. A presença deste módulo na estrutura mostrou-se bastante vantajosa, visto que a decisão de quando rodar a otimização influenciou, não apenas na performance computacional, mas também no resultado final da otimização.

A presença da estrutura D-RTO / MPC de forma modular, ou seja, em camadas distintas, não só provoca uma separação clara entre as tarefas das camadas de controle e otimização, facilitando a manutenção dos modelos, mas também fornece uma maior flexibilidade à estrutura, permitindo que a decisão de executar ou não a otimização seja feita no módulo do disparador.

Em todos os casos, a presença da camada de D-RTO na estrutura demonstrou melhor resultado em relação aos testes feitos utilizando no sistema de controle somente a camada MPC.

### *Sugestões para Trabalhos Futuros*

O uso da tecnologia da otimização dinâmica em tempo real (D-RTO) é o caminho natural das indústrias de processo para reduzir a quantidade de produtos fora de especificação e otimizar o lucro operacional diante das perturbações dinâmicas normalmente presentes no processo, tais como alterações de qualidade e quantidade de carga, transições frequentes consequência de alterações de programação de produção ou de receita de uma produção em batelada ou semi-batelada, dentre outros.

Esta tecnologia apresenta uma série de novos desafios frente à tecnologia de RTO (otimização estacionária). E uma série de questões novas ou novos pontos de vista aparecem quando se passa da otimização estacionária para a dinâmica.

Neste trabalho, foi dado um primeiro passo desenvolvendo uma estrutura de D-RTO tal qual a estrutura de RTO, porém inúmeros aspectos relacionados a essa estrutura ainda encontram-se em aberto. Algumas sugestões para evolução deste assunto são feitas a seguir:

- Utilização de um módulo de estimadores de estado do tipo filtro de *Kalman* estendido com restrições (CEKF) para prover condições iniciais consistentes para o algoritmo de otimização dinâmica; estimadores recursivos de parâmetros, também baseados em CEKF, para atualização dos modelos na camada de controle preditivo (MPC) e na camada de D-RTO.
- Utilização de um módulo com técnicas de análise estatística para discriminação de modelos.

- Testes em novos estudos de casos para contribuir ainda mais para consolidar esta tecnologia, porém sua aplicação a uma planta real é um desafio ainda a ser explorado.



## Referências Bibliográficas

- ABU-EL-ZEET, Z.H., ROBERTS P.D., BECERRA, V.M., 2002, “Enhancing Model Predictive Control Using Dynamic Data Reconciliation”, *AIChE Journal*, v.48 n.2, pp. 324-333.
- ALMEIDA, Nt. E., SECCHI A. R., 2006, *Dynamic Real-Time Optimization of a FCC Converter Unit*, In: ADCHEM 2006 - International Symposium on Advanced Control of Chemical Processes, Gramado, Brasil.
- ASCHER, U.; SPITERI R., 1995. *Collocation Software for Boundary Value Differential - Algebraic Equations*. Manual - University of British Columbia, Canada.
- BACKX, T., BOSGRA O., MARQUARDT W., 2000, “Integration of Model Predictive Control and Optimization of Processes”. In: *IFAC Symposium on Advanced Control of Chemical Processes*. v. 1., pp. 249-260.
- BADER, G., ASCHER, U. M., 1987, “A new basis implementation for mixed order boundary value ODE solver”. *SIAM J. Sci. Comput.*, v. 8, pp. 483-500.
- BARTLETT, R.A., BIEGLER L.T., 2003, “rsqp++ : An object-oriented framework for successive quadratic programming. In *Large-scale PDE-Constrained Optimization*”, *Lecture Notes in Computational Science and Engineering 30*, pp. 316, Berlin, Springer Verlag.

- BARTON, P., PANTELIDES, C., 1994, "Modeling of combined discrete/continuous processes", *AIChE Journal*, v. 40, pp. 966–979.
- BECERRA, V. M., ROBERTS, P. D., GRIFFITHS, G. W., 1998, "Novel Developments in Process Optimization Using Predictive Control". *J. Process Control*, v. 8, n. 2, pp. 117.
- BESL, H., KOSSMAN, W., CROWE, T. J., CARACOTSIOS, M., 1998, "Nontraditional Optimization for Isom Unit Improves Profits". *Oil Gas J.*, v. 96, n. 19, pp. 61.
- BETTS, J. T., 2001, *Practical Methods for Optimal Control Using Nonlinear Programming. Advances in Design and Control 3*, SIAM, Philadelphia, U.S.A.
- BINDER, T., WATZDORT, R.V., MARQUARDT, W., 1997, *Multiscale Perspectives on Process Simulation and Dynamic Optimization*. In: Technical Report LPT-1997-18, RWTH Aachen University of Technology.
- BINDER, T., BLANK, L., DAHMEN, W., MARQUARDT, W., 1999, *Adaptive Multiscale Approach to Real-Time Dynamic Optimization*. In: Technical Report LPT-1999-14. RWTH Aachen University of Technology.
- BIEGLER, L.T., 1984, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation". *Comput. Chem. Engng.*, v. 8, pp. 243 – 248.

- BIEGLER, L.T., NOCEDAL, J., SCHMID, C., 1995, “A reduced Hessian method for large-scale constrained optimization”. *SIAM J. Optimization*. v. 5, n. 2, pp. 314 – 347.
- BIEGLER L.T., 2000, “Efficient solution of dynamic optimization and NMPC problems”. In: Allgoewer, F., Zheng, A. (Eds.), *Nonlinear model predictive control*, pp. 219–245, Basel: Birkhaeuser.
- BIEGLER, L.T., CERVANTES, A.M., WÄCHTER, A., 2002, “Advances in simultaneous strategies for dynamic process optimization”, *Chemical Engineering Science*, v. 57, n. 4, pp. 575-593.
- BELLMAN, R. E., 1957, *Dynamic Programming*, Princeton University Press, New Jersey.
- BLISS, G.A., 1946, *Lectures on the Calculus of Variations*, University of Chicago Press, Chicago.
- BOCK, H.G., 1983, “Recent advances in parameter identification techniques for ODE”. In: Deuhard P., Hairer E. (eds). *Numerical Treatment of Inverse Problems in Differential and Integral Equations*. Birkhäuser, Boston.
- BOCK, H.G., PLITT, K.J., 1984, “A multiple shooting algorithm for direct solution of optimal control problems”. *International Federation of Automatic Control*, 9th World Congress, Budapest.

- BOCK, H.G., EICH, E., SCHLODER, J.P., 1988, "Numerical solution of constrained least squares boundary value problems in differential-algebraic equations". In: Strehmel K. (ed) *Numerical Treatment of Differential Equations*. Teubner, Leipzig.
- BOCK, H.G., BAUER, I., LEINEWEBER, D.B., SCHLÖDER, J.P., 1999, "Direct multiple shooting methods for control and optimization of DAE in chemical engineering". In: Keil F., Mackens W., Voss H., Werther J., (eds), *Scientific Computing in Chemical Engineering II*, v.2, pp. 2-18, Berlin, Springer.
- BRENAN, K.E, PETZOLD, L.R., 1989, "The Numerical Solution of Higher Index Differential/Algebraic Equations by Implicit Methods", *SIAM Journal on Numerical Analysis*, v. 26, n.4, pp. 976-996.
- BRENDEL, M., OLDENBURG, J., SCHLEGEL, M., STOCKMANN, K., 2004. *DyOS 2.1 User's Guide*, Software Manual - LPT-pro-2002-14.
- CERVANTES, A., BIEGLER, L.T., 1998, "Large-scale DAE optimization using simultaneous nonlinear programming formulations". *AIChE Journal*, v.44, pp. 1038-1050.
- CERVANTES, A. M., TONELLI, S., BRANDOLIN, A., BANDONI, J. A., BIEGLER, L.T., 2000a, "Large-scale dynamic optimization for grade transitions in a low density polyethylene plant". *Computers and Chemical Engineering*, v.24, pp. 983-989.

- CERVANTES, A. M., WACHTER, A., TUTUNCU, R., BIEGLER, L. T., 2000b, “A reduced space interior point strategy for optimization of differential algebraic systems”. *Computers and Chemical Engineering*, v.24, pp. 39–51.
- CHEN, H., KREMLING, A., ALLGOWER, F., 1995, “Nonlinear Predictive Control of a Benchmark CSTR”, *Proc. of 3<sup>rd</sup> ECC*, Rome, Italy, pp. 3247-3252.
- CHENG, J., ZAFIRIOU, E., 2000, “Robust Model-Based Iterative Feedback Optimization of Steady-State Plant Operations”, *Ind. Eng. Chem. Res.* v.39, pp. 4215.
- ČIŽNIAR, M., FIKAR, M., LATIFI, M.A., 2006, *MATLAB Dynamic Optimisation Code DYNOPT*, User's guide, Technical Report, KIRP FCHPT STU, Bratislava.
- COSTA, E. F., VIEIRA, R.C., SECCHI, A.R., BISCAIA, E.C., 2003, “Dynamic simulation of high-index models of batch distillation processes”, *Lat. Am. Appl. Res.*, v.33, n.2, p.155-160.
- CUTHRELL, J. E., BIEGLER L. T., 1987, “On the optimization of differential-algebraic process systems”. *AIChE Journal*, v.33, pp. 1257–1270.
- CUTHRELL, J. E., 1986, *On the Optimization of Differential-Algebraic Systems of Equations in Chemical Engineering*. Tese D.Sc., Carnegie Mellon University, Pittsburgh.

- CUTHRELL, J. E., BIEGLER, L. T., 1989, "Simultaneous-optimization and solution methods for batch reactor control profiles", *AIChE Journal*, v.13, pp. 49-62.
- DOWNS, J. J., VOGEL, E. F., 1993, "A plant-wide industrial process control problem". *Comp. Chem. Eng.*, v.17, n.3, pp.245-255.
- DUARTE, C. R. M., FIGUEIREDO, L. C., CORRÊA, M. V., 2006, "Utilização do Matlab® no Ensino da Tecnologia OPC Aplicada a Controle de Processos". *Anais do XVI Congresso Brasileiro de Automática*, Salvador.
- ENGELL, S., 2006, "Feedback Control for Optimal Process Operation", *International Symposium on Advanced Control of Chemical Processes*, Gramado, Brazil.
- ENGELL, S., KLATT, K.U., 1993, "Nonlinear Control of a Non-Minimum-Phase CSTR", *Proc. Of Control Conference*, Los Angeles, pp. 2041-2045.
- FEEHERY, W. F., BARTON P.I., 1998, "Dynamic optimization with state variable path constraints". *Comp. Chem. Eng.*, v.22, pp.1241-1256.
- GATH, P.F., 2002, *CAMTOS - A Software Suite Combining Direct and Indirect Trajectory Optimization Methods*. Tese D.Sc., Universität Stuttgart.
- gPROMS User Guide Version 2.0, 2001. *Process Systems Enterprise Ltd.*, London.
- GRIEWANK, A., KOWARZ, A., UTKE, J., VOGEL, O., WALTHER, A., 2004, "ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++", *Reference Manual-Version 1.9.0*, pp. 131-167, Argonne National Lab.

- JOCKENHÖVEL, T., BIEGLER L.T., WÄCHTER A., 2003, “Dynamic optimization of the Tennessee Eastman process using the OptControlCentre”. *Computers & Chemical Engineering*, v. 27, n.11, pp.1513-1531.
- KADAM, J.V., MARQUARDT W., 2004, *Sensitivity-based Solution Updates in Closed-loop Dynamic Optimization*. In: Technical Report LPT–2004–25, RWTH Aachen.
- KADAM, J.V., MARQUARDT, W., SCHLEGEL M., BACKX T., BOSGRA O.H., BROUWER, P.J., DÜNNEBIER G., HESSEM, D.V., TIAGOUNOV, A., WOLF S. D., 2003, *Towards Integrated Dynamic Real-time optimization and Control of Industrial Processes*. In: Technical Report LPT–2003–02, RWTH Aachen.
- KADAM, J.V., SCHLEGEL M., MARQUARDT W., TOUSAIN, R.L., HESSEM, D. H. V., BERG J.v.D., BOSGRA, O.H., 2001, *A Two-Level Strategy of Integrated Dynamic Optimization and Control of Industrial Processes – a Case Study*. In: Technical Report LPT–2001–29, RWTH Aachen.
- KAMIEN, M.I, SCHWARTZ, N.L., 1991, *Dynamic Optimization. The Calculus of Variations and Optimal Control in Economics and Management*, 2 ed., Elsevier Science.
- LANG, Y.D., BIEGLER, L. T., 2005, “A Software Environment for Simultaneous Dynamic Optimization”, *submitted to Comp. Chem. Engng.*

- LOGSDON, J.S., BIEGLER, L.T., 1989, "Accurate solution of differential-algebraic optimization problems". *Chem. Eng. Sci.*, v.28, pp.1628–1639.
- LEINWEBER, D.B., 1996, *The theory of MUSCOD in a nutshell*. Ms. Diss., University of Heidelberg.
- LEINWEBER, D.B., BOCK, H.G., SCHLÖDER, J.P., GALLITZENDÖFER, J.V., SCHÄFER, A., JANSOHN, P., 1997, *A boundary value problem approach to the optimization of chemical processes described by DAE models*. IWR-Preprint 97-14, University of Heidelberg.
- LEINWEBER, D.B., *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*. 1999, Tese D.Sc., University of Heidelberg.
- LI, W.C., BIEGLER, L.T., 1988, "Process control strategies for constrained nonlinear systems". *Ind. Eng. Chem. Res.*, v. 27, n.8, pp. 1421 – 1433.
- LOCKE, M.H., WESTERBERG, A.W., EDAHL, R.H., 1983, "Improved successive quadratic programming optimization algorithm for engineering design problems". *AIChE Journal*, v. 29, pp. 871 – 874.
- LOGSDON, J.S., BIEGLER, L.T., 1989, "Accurate solution of differential-algebraic optimization problems". *Chem. Eng. Sci.*, v.28, pp.1628–1639.



- LONGHI, L.G.S., 2001, *Solução do Problema de Controle  $H_\infty$  Não-Linear*. Tese D.Sc. - COPPE/UFRJ, Engenharia Química.
- LUUS, R., 1990, “Application of Dynamic Programming to High-Dimensional Nonlinear Optimal Control Problems”. *Int. J. Control*, v.52, n.1, pp. 239 – 250.
- LUUS, R., ROSEN O., 1991, “Application of Dynamic Programming to Final State Constrained Optimal Control Problems”. *Ind. Eng. Chem. Res.*, v.30, n.7, pp. 1525 – 1530.
- MILETIC, I.P., MARLIN, T.E., 1998, “On-line statistical results analysis in real-time operations optimization”. *Ind. Eng. Chem. Res.*, v.37, pp. 3670 – 3684.
- MIZOGUCHI, A., MARLIN, T., HRYMAK, A., 1995, “Operations Optimization and Control Design for a Petroleum Distillation Process”. *Can. J. Chem. Eng.*, v.73, pp. 896.
- NATH, R., ALZEIN, Z., 2000, “On-line dynamic optimization of olefin plants”. *Comput. Chem. Eng.*, v.24, pp.533.
- NEWMAN, C.P., SEN, A., 1974, “Weighted Residual Methods in Optimal Control”. *IEEE Transactions on Automatic Control*, pp. 67 – 69.
- PANTELIDES, C.C, VASSILIADIS, V.S., SARGENT, R.W.H., 1994, “Optimal control of multistage systems described by high-index differential-algebraic

equations". In: Bulirsch, R., Kraft, D. (eds.), *Computational Optimal Control*, Birkhäuser, Basel, Germany, pp. 177–191.

PETZOLD, L., REN, Y., MALY, T., 1997, "Regularization of higher-index differential-algebraic equations with rank-deficient constraints". *SIAM J. Sci. Comput.*, v.18, pp. 753–774.

PETZOLD, L.R., 1989, "DASSL code, version 1989" ,*Computing and Mathematics Research Division*, Lawrence Livermore National Laboratory, L316, PO Box 808, Livermore, CA 94559.

PLITT, K.J., 1988, *Ein superlinear konvergentes Mehrziel verfahren zur direkten Berechnung beschränkter optimaler Steuerungen*. Diplomarbeit, Universität Bonn.

POLLARD, G.P., SARGENT, R.W.H., 1970, "Off Line Computation of Optimum Controls for a Plate Distillation Column". *Automatica*, v. 6, n.6, pp. 59 – 76.

PONTRYAGIN, V., BOLTYANSKII, V., GAMKRELIDZE R., MISHCHENKO E., 1962, *The Mathematical Theory of Optimal Processes*. Interscience Publishers Inc., New York, NY.

PYTLAK, R., VINTER, R.B., 1996a, *A Feasible Directions Type Algorithm for Optimal Control Problems with State and Control Constraints: Convergence Analysis*. In: Technical report, Imperial College of Science, Technology and Medicine, London, England.

- PYTLAK, R., VINTER, R.B., 1996b, *A Feasible Directions Type Algorithm for Optimal Control Problems with State and Control Constraints: Implementation*. In: Technical report, Imperial College of Science, Technology and Medicine, London, England.
- QIN, S. J., BADGWELL, T. A., 1997, "An overview of industrial model predictive control technology". *AIChE Symp. Ser.*, v.316, n.93, pp.232.
- RAO, C. V., RAWLINGS, J. B., 1999, "Steady States and Constraints in Model Predictive Control". *AIChE J.*, v.45, n.6, pp.1266.
- RAO, C. V., RAWLINGS, J. B., LEE, J. H., 2001, "Constrained linear state estimation – a moving horizon approach". *Automatica*, v.37, n.10, pp. 1619–1628.
- RAO, C. V., RAWLINGS, J. B., MAYNE, D. Q., 2003. "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations". *IEEE Transactions on Automatic Control*, v.48, n.2, pp. 246-258.
- RENFRO, J.G., 1986, *Computational Studies in the Optimization of Systems Described by Differential/Algebraic Equations*. Tese D.Sc., University of Houston, University Park.
- RENFRO, J.G., 1987, MORSHEDI, A.M., ASBJORNSEN, O.A., 1987, "Simultaneous optimization and solution of systems described by differential algebraic equations", *Computer and Chemical Engineering*, v.11, n.5, pp. 503-517.

- ROTAVA, O., 1997, *Implementation of Linear and Nonlinear Optimal Control Techniques in a Carbon Dioxide Absorption Desorption Plant*. Tese D.Sc., Imperial College Science, London.
- SALAU, N.P.G., 2009, *Abordagem Sistemática para Construção e Sintonia de Estimadores de Estados Não-Lineares*. Tese de Doutorado, PPGEQ/UFRGS, Porto Alegre, RS.
- SANTOS, L.O., AFONSO, P., CASTRO, J., OLIVEIRA, N.M.C. de, BIEGLER, L.T., 2000, *On-line implementation of nonlinear MPC: Na experimental case study.*, In: ADCHEM 2000 – International Symposium on Advanced Control of Chemical Processes, v.2, pp. 731-736, Pisa
- SARGENT, R.W.H, SULLIVAN, G.R., 1977, *The Development of an Efficient Optimal Control Package*. In: Proc. of the 8th IFIP Conf. on Optimization Techniques Part 2.
- SCHLEGEL, M., BINDER, T., CRUSE, A., OLDENBURG, J., MARQUARDT, W., 2000, *ADOPT – a Software Tool for Adaptive Dynamic Optimization*. Technical Report LPT-pre-2000-16.
- SCHULZ, V.H, 1996, *Reduced SQP Methods for Large Scale Optimal Control Problems in DAE with Application to Path Planning Problems for Satellite Mounted Robots*. Tese D.Sc. Universitat Heidelberg.

- SEQUEIRA, S. E., GRAELLS, M., PUIGJANER, L., 2002, "Real Time Evolution for On-line Optimization of Continuous Processes", *Ind. Eng. Chem. Res.* v. 41, pp. 1815-1825.
- SERBAN, R., PETZOLD, L.R., 2001, "COOPT - A Software Package for Optimal Control of Large-Scale Differential-Algebraic Equation Systems", *J. Math. and Computers in Simulation*, v.56, n.2, pp. 187-204.
- SOARES, R.P., SECCHI, A.R., 2003, "EMSO: A new Environment for Modelling, Simulation and Optimisation", *ESCAPE-13*.
- SORENSEN, R, C., CUTLER, C. R., 1998, "LP integrates economics into dynamic matrix control". *Hydrocarbon Process Sep*, 57.
- STRYK, O.v., 1999, *User's Guide for DIRCOL Version2.1 – A Direct Collocation Method for the Numerical Solution of Optimal Control Problems*. Technische Universität Darmstadt.
- TANARTKIT, P., BIEGLER, L.T., 1995, "Stable decomposition for dynamic optimization". *Ind. Eng. Chem. Res.*, v.34, pp. 1253-1266.
- TANARTKIT, P., BIEGLER, L. T., 1996, "A nested, simultaneous approach for dynamic optimization problems – I". *Comput. Chem. Eng.* v.20, n.6/7, pp. 735-741.

- TRIERWEILER, J.O., FARINA, L.A., 2003, "RPN Tuning Strategy for Model Predictive Control", *Journal of Process Control*, pp. 591-598.
- TSANG, T.H., HIMMELBLAU, D.M., EDGAR, T.F., 1975, "Optimal control via collocation and nonlinear programming". *Int. J. Control*, v.21, n.5, pp. 763-768.
- TVRZSKA', M., ODLOAK, D., 1998, "One-Layer Real Time Optimization of LPG Production in the FCC Unit: Procedure, Advantages and Disadvantages". *Comput. Chem. Eng.*22, S191.
- VASANTHARAJAN, S., BIEGLER, L.T., 1988, "Large-scale decomposition for successive quadratic programming", *Computers chem. Engng.*, v.12, pp. 1087-1101.
- VASANTHARAJAN, S., BIEGLER, L.T., 1990, "Simultaneous strategies for optimization of differential algebraic systems with enforcement of error criteria", *Computers chem. Engng.*, v.14, pp. 1083 - 1100.
- VASSILIADIS, V.S., 1993, *Computational Solution of Dynamic Optimization Problems with General Differential-Algebraic Constraints*, Tese D.Sc., University of London, London, U.K.
- VASSILIADIS, V. S., SARGENT, R.W.H., PANTELIDES, C.C., 1994a, "Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints", *Ind. Eng. Chem. Res.*, v.33, n.9, pp. 2111-2122.

VASSILIADIS, V.S., SARGENT, R.W.H., PANTELIDES, C.C., 1994b, "Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints", *Ind. Eng. Chem. Res.*, v.33, n.9, pp. 2123-2133.

WÄCHTER, A., 2002, *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. Tese D.Sc., Carnegie Mellon University, Pittsburgh, PA, USA.

WÄCHTER, A., BIEGLER, L.T., 2005, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". *Math. Program., Ser. A*.

WHITE, D. C., 1998, "On-line Optimization: What Have We Learned?", *Hydrocarbon Process. Jun*, 55.

YING, C.-M., JOSEPH, B., 1999, "Performance and Stability Analysis of LP-MPC and QP-MPC Cascade Control Systems". *AIChE J.*, v.45, n.7, pp. 1521.

ZANIN, A.C., 2001, *Implementação Industrial de um Otimizador em Tempo Real*, D.Sc. Tese, EPUSP São Paulo.

# Apendice A - Código em Matlab da *S-Function* para o estudo de casos do reator CSTR

Abaixo segue descrito o modelo do reator CSTR com reação de Van der Vusse implementado na *S-Function* em linguagem de Matlab.

```

%-----
% function [sys,x0,str,ts] = SFNCCSTR(t,x,u,flag)
% description: planta do processo.
%-----
function [sys,x0,str,ts] = SFNCCSTR(t,x,u,flag)

%%%%%%%%%%
% DATA %
%%%%%%%%%%

%Parametros Comuns
pi = 3.141593;

A = evalin('base','A');
B = evalin('base','Bm');

DH1 = evalin('base','DH1');
DH2 = evalin('base','DH2');
DH3 = evalin('base','DH3');
cp = evalin('base','cp');
rho = evalin('base','rho');
k1 = evalin('base','k1');
k2 = evalin('base','k2');
k3 = evalin('base','k3');
E1 = evalin('base','E1');
E2 = evalin('base','E2');
E3 = evalin('base','E3');
Di = evalin('base','Di');
Area = evalin('base','Area');
H = evalin('base','H');
U = evalin('base','U');
cv = evalin('base','cv');
Vt = Area*H; %Volume total do reator
Atmax = Area + pi*Di*H; %area de transferencia máxima

switch flag,

%%%%%%%%%%
% Initialization %
%%%%%%%%%%
case 0,
    sizes = simsizes;

    sizes.NumContStates = 5; %Variaveis de estado (continuas).
    sizes.NumDiscStates = 0; %Variaveis de estado (discretas).

```



```

    sizes.NumOutputs      = 5; %Saidas da s-function.
    sizes.NumInputs       = 5; %Entradas p a s-function - F
    sizes.DirFeedthrough = 1; %Essa opcao permite q o output (flag
3) reconheca as entradas u.
    sizes.NumSampleTimes = 1; %Numero de tempo de amostragem - p
continuo (nesse caso) deve ser 1.

    sys = simsizes(sizes);

%Condições Iniciais
Ca0 = 5.5; % kmol/m^3
Cb0 = 0; % kmol/m^3
Cc0 = 0; % kmol/m^3
Cd0 = 0; % kmol/m^3
T0 = evalin('base','Ts'); % K

x0 = [Ca0 Cb0 Cc0 Cd0 T0];

str = [];
ts = [0 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Derivatives % Usado p integracao de modelos p variaveis de estado
continuas.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 1,

%+++++

    Ff = u(1);
    Caf = u(2);
    Cbf = u(3);
    Ccf = 0;
    Cdf = 0;
    Tf = u(4);
    Tw = u(5);

    h = H; % altura fixa
    Ca = x(1);
    Cb = x(2);
    Cc = x(3);
    Cd = x(4);
    T = x(5);

% Equacoes
    k1T = k1*exp(-E1/T);
    k2T = k2*exp(-E2/T);
    k3T = k3*exp(-E3/T);
    V = Area * h;
    tau = V / Ff;
    r1 = k1T*Ca;
    r2 = k2T*Cb;
    r3 = k3T*Ca*Ca;
    At = Area + pi*Di*h;

    qr = - V * ( DH1*r1 + DH2*r2 + DH3*r3 );
    q = U * At * (T - Tw);

    sys(1) = (1/tau) * ( Caf - Ca ) - r1 - r3;

```

```

sys(2) = (1/tau) * ( Cbf - Cb ) + r1 - r2;
sys(3) = (1/tau) * ( Ccf - Cc ) + r2;
sys(4) = (1/tau) * ( Cdf - Cd ) + r3/2;
sys(5) = (1/tau) * ( Tf - T ) + ( qr - q ) / (rho*V*cp) ;

%+++++
%*****
% Update % Usado qdo se tem variaveis de estado discretas.
%*****
case 2,
    sys = [];
%
%*****
% Outputs % Saidas da s-function
%*****
case 3,
    sys(1) = x(1); %Ca
    sys(2) = x(2); %Cb
    sys(3) = x(3); %Cc
    sys(4) = x(4); %Cd
    sys(5) = x(5); %T

    assignin( 'base','CurrentTime', t);

%*****
% GetTimeOfNextVarHit % Mudanca dos tempos de amostragem, qdo os
mesmos sao variaveis
%*****
case 4,
    sys = [];
%*****
% Terminate %
%*****
case 9,
    sys = [];

%*****
% Unexpected flags %
%*****
otherwise
    sys = [];
    error(['Unhandled flag = ',num2str(flag)]);
end

```

## Apêndice B- Código em Matlab para o Módulo OPC

```
%-----  
% function outputData = opc_module(u)  
% description: Faz a leitura e a escrita dos dados na planta via OPC.  
%-----  
function outputData = opc_module(u)  
  
newFvol = u(1);  
newTw = u(5);  
  
%Escreve os dados no servidor para as variáveis manipuladas  
write(Feed_vol, newFvol);  
write(Reactor_Tw, newTw);  
  
%Leitura dos dados no servidor para informações da planta  
outCa = read(Product_Ca);  
outCb = read(Product_Cb);  
outCc = read(Product_Cc);  
outCd = read(Product_Cd);  
outT = read(Product_T);  
  
y(1) = outCa.Value;  
y(2) = outCb.Value;  
y(3) = outCc.Value;  
y(4) = outCd.Value;  
y(5) = outT.Value;  
  
outputData = y;
```

O método descrito abaixo deve ser executado na etapa de inicialização:

```
%-----  
% function enable = [G, Feed_Vol, Feed_Ca, Feed_Cb, Feed_Cc, Feed_Cd,  
% Feed_T, Reactor_Tw] = opc_init()  
% description: Inicializa as variáveis de comunicação com o servidor  
% OPC.  
%-----  
function [G, Feed_Vol, Feed_Ca, Feed_Cb, Feed_Cc, Feed_Cd, Feed_T,  
Reactor_Tw] = opc_init()  
  
Obj = opcda('localhost', 'DLLTestSvr');  
connect(Obj);  
G=addgroup(Obj, 'Gr');  
  
assignin('base', 'G', G);  
  
Feed_Vol = additem(G, 'Feed.Fvol');  
Feed_Ca = additem(G, 'Feed.Ca');  
Feed_Cb = additem(G, 'Feed.Cb');
```

```
Feed_Cc = additem(G, 'Feed.Cc');
Feed_Cd = additem(G, 'Feed.Cd');
Feed_T = additem(G, 'Feed.T');
Reactor_Tw = additem(G, 'Reactor.Tw');

assignin('base', 'Feed_Vol', Feed_Vol);
assignin('base', 'Feed_Ca', Feed_Ca);
assignin('base', 'Feed_Cb', Feed_Cb);
assignin('base', 'Feed_Cc', Feed_Cc);
assignin('base', 'Feed_Cd', Feed_Cd);
assignin('base', 'Feed_T', Feed_T);
assignin('base', 'Reactor_Tw', Reactor_Tw);

end
```

# Apêndice C - Código do modelo do reator CSTR implementado no

## EMSO

Classe das correntes materiais:

```
1  ##-----*
2  * EMSO Model Library (EML) Copyright (C) 2004 - 2009 ALSOC.
3
4  * This LIBRARY is free software; you can distribute it and/or modify
5  * it under the terms of the ALSOC FREE LICENSE as available at
6  * http://www.enq.ufrgs.br/alsoc.
7
8  * EMSO Copyright (C) 2004 - 2009 ALSOC, original code
9  * from http://www.rps.eng.br Copyright (C) 2002-2004.
10 * All rights reserved.
11
12 * EMSO is distributed under the terms of the ALSOC LICENSE as
13 * available at http://www.enq.ufrgs.br/alsoc.
14
15 ##-----*
16 * CSRT Non Isothermic - Van der Vusse Reaction
17 ##-----*
18 * Author: Otto Indio do Brasil Magalhaes
19 * $Id: cstr_vdv_noniso.mso 2009-09-19 20:49:17Z oibm $
20 ##-----*
21
22 using "simple_streams";
23
24 Model stream_conc as MaterialStream
25
26 ATTRIBUTES
27     Pallette      = false;
28     Brief         = "Material stream with molar concentration.";
29     Info          =
30     "==" Contents ==
31     *Ca: Molar Concentration of Component A
32     *Cb: Molar Concentration of Component B
33     *Cc: Molar Concentration of Component C
34     *Cd: Molar Concentration of Component D
35     ";
36
37 VARIABLES
38
39     Ca as conc_mol (Brief = "Molar concentration component A", Lower = -10000,
40     Cb as conc_mol (Brief = "Molar concentration component B", Lower = -10000,
41     Cc as conc_mol (Brief = "Molar concentration component C", Lower = -10000,
42     Cd as conc_mol (Brief = "Molar concentration component D", Lower = -10000,
43
44 end
45
```

Classe do modelo CSTR com reação de Van der Vusse não Isotérmico:

```

cstr_vdv_noniso.mso
Model cstr_vdv_noniso
92
93
94 ATTRIBUTES
95   Palette      = true;
96   Icon         = "icon/cstr";
97   Brief        = "Model of a cstr with Van Der Vusse Reaction.";
98   Info         =
99   "==" Assumptions ==
100  * the reactor is a well mixed element;
101  * non isothermic reactor
102  * only designed for Van Der Vusse Reaction;
103
104  == Specify ==
105  * the Inlet stream;
106
107  == Setting Parameters ==
108  *Reaction rate constant (k1);
109  *Reaction rate constant (k2);
110  *Reaction rate constant (k3);
111  *Reactor Area (A);
112  *Valve constant (Cv);
113
114  == Initial Conditions ==
115  * the reactor initial level (h)
116  * the components initial molar concentration inside reactor (Ca,Cb,Cc,Cd);
117  ";
118
119  PARAMETERS
120
121  k1  as frequency      (Brief="Reaction rate constant", DisplayUnit='1/min', Symbo
122  k2  as frequency      (Brief="Reaction rate constant", DisplayUnit='1/min', Symbo
123  k3  as positive       (Brief="Reaction rate constant", Unit='1/mol/min', Symbol =
124  E1  as temperature    (Brief="Activation energy", Symbol = "E_1");
125  E2  as temperature    (Brief="Activation energy", Symbol = "E_2");
126  E3  as temperature    (Brief="Activation energy", Symbol = "E_3");
127  Cv  as positive       (Brief = "Valve Constant", Default=1,Lower=0,Upper=100, Uni
128  A  as area;
129  Hmax as length        (Brief = "Reactor max length");
130  D  as length          (Brief = "Reactor diameter");
131  rho as dens_mass      (DisplayUnit='kg/m^3');
132  cp  as cp_mass        (DisplayUnit='kJ/kg/K');
133  U  as heat_trans_coeff (DisplayUnit='kW/m^2/K');
134  H1  as heat_reaction  (DisplayUnit='kJ/kmol');
135  H2  as heat_reaction  (DisplayUnit='kJ/kmol');
136  H3  as heat_reaction  (DisplayUnit='kJ/kmol');
137  pi  as Real           (Default = 3.141593);
138
139
140  VARIABLES
141  V  as volume          (Brief = "Reactor volume");
142  Ca  as conc_mol       (Brief = "Molar concentration component A", Lower =
143  Cb  as conc_mol       (Brief = "Molar concentration component B", Lower =
144  Cc  as conc_mol       (Brief = "Molar concentration component C", Lower =
145  Cd  as conc_mol       (Brief = "Molar concentration component D", Lower =
146  h  as length          (Brief = "Reactor length");
147  tau as time_min       (Brief = "Reactor mean residence time", Symbol =
148  r1  as reaction_mol   (Brief = "Reaction rate", Symbol = "r_1");
149  r2  as reaction_mol   (Brief = "Reaction rate", Symbol = "r_2");
150  r3  as reaction_mol   (Brief = "Reaction rate", Symbol = "r_3");
151  At  as area;
152  T  as temperature;
153  Tw  as temperature;
154  q  as heat_rate       (DisplayUnit='kJ/s');
155  qr  as heat_rate      (DisplayUnit='kJ/s');
156  x  as fraction;
157
158  SUBMODELS
159
160  in  Inlet  as stream_conc (Brief="Inlet stream",PosX=0, PosY=0, Symbol =
161  out  Outlet as stream_conc (Brief="Outlet stream", PosX=1, PosY=1, Symbol = "A
162
163
164  SET
165  A = pi * DA2 / 4;
166

```

```

167 EQUATIONS
168
169 "Overall Mass Balance"
170   diff(V) = Inlet.Fvol - Outlet.Fvol;
171
172 "Reaction Rate - First Reaction"
173   r1 = k1*exp(-E1/Outlet.T)*Ca;
174
175 "Reaction Rate - Second Reaction"
176   r2 = k2*exp(-E2/Outlet.T)*Cb;
177
178 "Reaction Rate - Third Reaction"
179   r3 = k3*exp(-E3/Outlet.T)*Ca*Ca;
180
181 "A Component Mass Balance"
182   tau*diff(Ca) = -tau*r1 - tau*r3 + Inlet.Ca - Outlet.Ca;
183
184 "B Component Mass Balance"
185   tau*diff(Cb) = tau*r1 - tau*r2 + Inlet.Cb - Outlet.Cb;
186
187 "C Component Mass Balance"
188   tau*diff(Cc) = tau*r2 + Inlet.Cc - Outlet.Cc;
189
190 "D Component Mass Balance"
191   tau*diff(Cd) = tau*(1/2)*r3 + Inlet.Cd - Outlet.Cd;
192
193 "Well Mixed - A component"
194   Outlet.Ca = Ca;
195
196 "Well Mixed - B component"
197   Outlet.Cb = Cb;
198
199 "Well Mixed - C component"
200   Outlet.Cc = Cc;
201
202 "Well Mixed - D component"
203   Outlet.Cd = Cd;
204
205 "Well Mixed - Temperature"
206   Outlet.T = T;
207
208 "Residence Time"
209   tau = V/Inlet.Fvol;
210
211 "Energy Balance"
212   ro * V * cp * diff(T) = Inlet.Fvol * ro * cp * (Inlet.T - T) + qr - q;
213
214 "Heat Transfer Rate"
215   q = U * At * (T - Tw);
216
217 "Reaction Heat Rate"
218   qr = -V * (H1*r1 + H2*r2 + H3*r3);
219
220 "Reactor Volume"
221   V = A * h;
222
223 "Mechanical equilibrium"
224   Outlet.P = Inlet.P;
225
226
227 if (h >= Hmax) then
228   "Heat Transfer Area"
229   At = A + pi*D*Hmax;
230
231   "Valve Equation"
232   Outlet.Fvol = Inlet.Fvol;
233
234 else
235   "Heat Transfer Area"
236   At = A + pi*D*h;
237
238   "Valve Equation"
239   Outlet.Fvol = x * Cv * sqrt(h);
240
241 end
242
243
244
245 end
246
247

```