

Universidade Federal de Campina Grande
Unidade Acadêmica de Engenharia Química
Programa de Pós-Graduação em Engenharia Química

**IMPLEMENTAÇÃO DA METODOLOGIA DE *SELF-OPTIMIZING*
CONTROL USANDO METAMODELOS *KRIGING* EM LINGUAGEM
*PYTHON***

Marcílio Máximo da Silva

Campina Grande, Paraíba, Brasil
2023

Universidade Federal de Campina Grande
Unidade Acadêmica de Engenharia Química
Programa de Pós-Graduação em Engenharia Química

**IMPLEMENTAÇÃO DA METODOLOGIA DE *SELF-OPTIMIZING*
CONTROL USANDO METAMODELOS *KRIGING* EM LINGUAGEM
*PYTHON***

Marcílio Máximo da Silva

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal de Campina Grande como um dos requisitos para obtenção do título de Doutor em Engenharia Química.

Orientadores: Dr. Heleno Bispo da Silva Júnior
Dr. Antônio Tavernard Pereira da Silva Neto

Campina Grande, Paraíba, Brasil
2023

S586i

Silva, Marcílio Máximo da.

Implementação da metodologia de *self-optimizing control* usando metamodelos *kriging* em linguagem *Python* / Marcílio Máximo da Silva. - Campina Grande, 2023.

77 f. : il. color.

Tese (Doutorado em Engenharia Química) - Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia, 2023.

"Orientação: Prof. Dr. Heleno Bispo da Silva Júnior, Prof. Dr. Antônio Tavernard Pereira da Silva Neto"

Referências.

1. Self-optimizing control. 2. Metamodelos. 3. Kriging. 4. Python. I. Silva Júnior, Heleno Bispo da. II. Silva Neto, Antônio Tavernard Pereira da. III. Título.

CDU 66.011(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA ITAPUANA SOARES DIAS GONÇALVES CRB-15/093

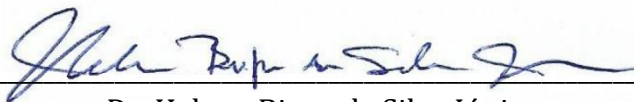
Marcílio Máximo da Silva

**IMPLEMENTAÇÃO DA METODOLOGIA DE *SELF-OPTIMIZING CONTROL*
USANDO METAMODELOS KRIGING EM LIGUAGEM PYTHON**

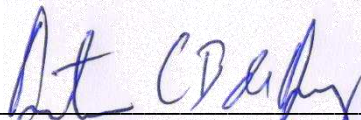
Tese de Doutorado aprovada em: **14/09/2023**




Dr. Antônio Tavernard Pereira da Silva Neto
Orientador - UFCG



Dr. Heleno Bispo da Silva Júnior
Orientador - UFCG



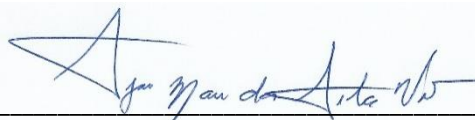
Dr. Antônio Carlos Brandão de Araújo
Examinador Interno - UFCG



Dr. José Jailson Nicácio Alves
Examinador Interno - UFCG



Dr. Sidinei Kleber da Silva
Examinador Externo - UFCG



Dr. Sérgio Mauro da Silva Neiro
Examinador Externo - UFU

Campina Grande, Paraíba, Brasil
2023

Dedico essa Tese de Doutorado para minha querida esposa Mariana e os meus
filhos Gael Máximo e Liam Máximo

AGRADECIMENTOS

À Deus pelo dom da vida.

À minha amada esposa Mariana, por todo amor, apoio, compreensão e incentivo durante todos esses longos anos de Doutorado. Essa tese só foi desenvolvida porque você não me deixou desistir, você estava do meu lado quando me faltou motivação e me fez acreditar que eu era capaz de completar mais essa etapa da minha vida acadêmica.

Aos meus filhos, que neste momento que eu escrevo não têm ideia do que essa tese é, mas eu irei registrar aqui que vocês, Gael e Liam, sempre serão as pessoas que me motivam para ser melhor a cada dia. Meu amor por vocês transcende esse mundo passageiro.

Aos meus pais, Manoel Máximo e Josefa Maria, minhas referências de vida, por terem aceitado o chamado de Deus para me dar o dom da vida, e por terem assumido a missão de me educar com virtudes que hoje carrego comigo e jamais serão esquecidas.

Ao meu irmão, Marlon, que mesmo longe fisicamente, sei que está sempre comemorando minhas conquistas.

Ao Professor Heleno, pela orientação ao longo de muitos anos. Obrigado por todo aprendizado compartilhado no mestrado, nos projetos, no desenvolvimento desta tese, na escrita de artigos, na manutenção dos laboratórios. Você transmite valores que vão além do conhecimento acadêmico. Você eleva os alunos a um nível profissional que não imaginamos que podemos chegar. Deixo aqui minha admiração e gratidão por tudo.

Ao Professor Tavernard, pela orientação e conhecimento compartilhado. Em uma das primeiras disciplinas que fui seu aluno na graduação em Engenharia Química você despertou a curiosidade em aprender programação e também a seguir em frente no curso. Tenho enorme admiração pela sua genialidade e gratidão por todo ensinamento que me transmitiu.

Ao engenheiro Herbert Teixeira, pelas discussões que tivemos ao longo do desenvolvimento dessa tese, por todo conhecimento compartilhado. Tenho enorme admiração pela sua humildade e disponibilidade em sempre contribuir para expansão do conhecimento através de situações reais da indústria.

Aos meus amigos do IFCE *campus* Quixadá, pelo apoio e incentivo durante o desenvolvimento deste trabalho.

Aos colegas do Iridium – UFCG que se mostraram disponíveis para contribuir com o desenvolvimento dessa tese.

A todos os meus amigos do Programa de Pós-Graduação em Engenharia Química – UFCG.

A cada pessoa que, a sua maneira, contribuiu com este trabalho.

“Trabalhe, Sirva, Seja Forte e Não Encha o Saco!”

(Dr. Italo Marsili)

RESUMO

Um dos principais objetivos da engenharia é operar processos de forma ótima ou próxima ao ponto ótimo, buscando maximizar ou minimizar uma função objetivo que pode levar em consideração restrições operacionais, ambientais e/ou econômicas. A garantia de operação ótima só é possível por meio de uma estrutura de controle bem projetada para manter o processo dentro das especificações. Neste trabalho, propomos um procedimento sistemático programado em linguagem *Python*, utilizando o ambiente de desenvolvimento *Jupyter Notebook*, para a metamodelagem, otimização e aplicação da metodologia de *self-optimizing control* (SOC). Esse procedimento foi desenvolvido com o objetivo de otimizar a sequência de implementação da metodologia de *self-optimizing control* para processos simulados usando o *Aspen Hysys*. Para análise e validação de sua funcionalidade o procedimento é aplicado para dois estudos de caso. No primeiro, um modelo desenvolvido em *Python* para um reator CSTR no qual ocorre uma reação reversível e exotérmica é analisado para determinação da variável controlada que resulta na menor perda. Os resultados obtidos são comparados com resultados da literatura, e todos são validados numericamente. No segundo estudo de caso, o procedimento é aplicado para analisar uma coluna de destilação de alta pureza modelada no simulador de processos *Aspen Hysys* para determinar os melhores pares de controle. Os resultados obtidos mostram coerência em termos do processo, validando assim a aplicabilidade do procedimento. A metodologia de *self-optimizing control* é uma abordagem promissora para otimizar a operação de processos industriais. Ao utilizar metamodelos, como o *Kriging*, é possível reduzir o esforço computacional necessário para realizar a otimização. Em uma das etapas mais importantes da metodologia SOC, modificações matemáticas foram realizadas para obtenção dos metamodelos. Neste trabalho mostramos o ganho computacional e de tempo através da utilização de regressores de segunda ordem para a estimativa das Hessianas. Por fim, o trabalho apresenta contribuições valiosas para a área de controle de processos, fornecendo uma ferramenta automatizada para a aplicação da metodologia SOC. Os resultados obtidos mostram que a ferramenta é eficaz na determinação da variável controlada em um reator CSTR e na identificação dos melhores pares de controle para uma coluna de destilação de alta pureza. Além disso, a validação numérica dos resultados reforça a confiabilidade e a utilidade da ferramenta proposta.

PALAVRAS-CHAVE: *Self-optimizing control*, Metamodelos, *Kriging*, *Python*

ABSTRACT

One of the primary objectives of engineering is to operate processes optimally or near the optimum point, aiming to maximize or minimize an objective function that may take into account operational, environmental, and/or economic constraints. Ensuring optimal operation is only possible through a well-designed control framework to maintain the process within specifications. In this work, we propose a systematic procedure programmed in the Python language, utilizing the Jupyter Notebook development environment, for metamodeling, optimization, and application of the self-optimizing control (SOC) methodology. This procedure was developed with the goal of optimizing the implementation sequence of the self-optimizing control methodology for simulated processes using Aspen Hysys. For analysis and validation of its functionality, the procedure is applied to two case studies. In the first case, a Python-developed model for a reversible and exothermic reaction in a CSTR reactor is analyzed to determine the controlled variable resulting in the least loss. The obtained results are compared with literature results, and all outcomes are numerically validated. In the second case study, the procedure is applied to analyze a high-purity distillation column modeled in the Aspen Hysys process simulator to determine the best control pairs. The obtained results exhibit coherence in terms of the process, thus validating the applicability of the procedure. The self-optimizing control methodology presents a promising approach for optimizing the operation of industrial processes. By employing metamodels like Kriging, it is possible to reduce the computational effort needed for optimization. In one of the most crucial steps of the SOC methodology, mathematical modifications were carried out to obtain metamodels. In this work, we demonstrate the computational and time gains achieved through the utilization of second-order regressors for Hessian estimation. Lastly, the study provides valuable contributions to the field of process control, furnishing an automated tool for applying the SOC methodology. The achieved results demonstrate the tool's effectiveness in determining the controlled variable in a CSTR reactor and identifying the optimal control pairs for a high-purity distillation column. Moreover, the numerical validation of results reinforces the reliability and utility of the proposed tool.

KEYWORDS: Self-optimizing control, Metamodels, Kriging, Python

LISTA DE FIGURAS

Figura 1. Hierarquia de uma estrutura de controle. (Fonte: Adaptado de Jäschke et al., 2017)	22
Figura 2. Código de comunicação entre o <i>Python</i> e o <i>Aspen Hysys</i>	47
Figura 3. Fluxograma para o algoritmo de planejamento metamodelagem e otimização	51
Figura 4. Fluxograma para o algoritmo de seleção das variáveis controladas	52
Figure 5. Ilustração do CSTR	54
Figure 6. Validação dos metamodelos para as variáveis de saída do CSTR. (a) Concentração do reagente A; (b) Concentração do reagente B; (c) Temperatura do reator; (d) Temperatura da corrente de entrada do reator	59
Figure 7. Validação dos metamodelos para a função objetivo	60
Figure 8. Função objetivo <i>versus</i> variável manipulada (T_i) para o caso nominal e diferentes distúrbios, os quais: $C_{Ai} = 0,7 \text{ mol.L}^{-1}$, $C_{Ai} = 1.3 \text{ mol.L}^{-1}$, $C_{Bi} = 0,3 \text{ mol.L}^{-1}$. A linha vertical em verde representa o ponto ótimo para o caso nominal.....	61
Figure 9. Curvas de nível para função objetivo em função das variáveis manipuladas. A restrição x_D é representada pela curva em vermelho	66

LISTA DE TABELAS

Tabela 1. Dados nominais para o CSTR	54
Tabela 2. Valores ótimos para diferentes distúrbios aplicados para os metamodelos do CSTR	60
Tabela 3. Comparação das perdas para os casos mais promissores para controle do reator CSTR	63
Tabela 4. Variáveis manipuladas, distúrbios e medidas e suas respectivas <i>tags</i> de automação no <i>Aspen Hysys</i>	64
Tabela 5. Resultados dos metamodelos e <i>Aspen Hysys</i> para o ponto ótimo	67
Tabela 6. Resultados das candidatas a variáveis controladas em ordem crescente das perdas	69

SUMÁRIO

RESUMO	9
ABSTRACT	10
LISTA DE FIGURAS.....	11
LISTA DE TABELAS.....	12
1. INTRODUÇÃO	14
1.1. PUBLICAÇÕES	15
2. OBJETIVOS	16
2.1. OBJETIVO GERAL	16
2.2. OBJETIVOS ESPECÍFICOS.....	16
3. FUNDAMENTAÇÃO TEÓRICA.....	17
3.1. <i>PLANTWIDE CONTROL</i>	17
3.2. <i>SELF-OPTIMIZING CONTROL</i>	19
3.3. DEFININDO O PROBLEMA DE OTIMIZAÇÃO	21
3.4. IMPLEMENTAÇÃO DO PROBLEMA DE OTIMIZAÇÃO	23
3.5. SELEÇÃO DAS VARIÁVEIS CONTROLADAS	25
3.5.1. Método da Força Bruta ou Busca Exaustiva	25
3.5.2. Método Exato Local.....	27
3.6. APLICAÇÃO DA METODOLOGIA DE <i>SELF-OPTIMIZING CONTROL</i>	39
3.7. METAMODELOS: <i>KRIGING</i>	41
3.8. OTIMIZAÇÃO: <i>INTERIOR POINT OPTIMIZER (IPOPT)</i>	43
5. IMPLEMENTAÇÃO DA METODOLOGIA	47
5.1. PLANEJAMENTO, METAMODELAGEM E OTIMIZAÇÃO.....	48
5.2. SELEÇÃO DAS VARIÁVEIS CONTROLADAS	52
6. ESTUDOS DE CASO	54
6.1. ESTUDO DE CASO 01: CSTR COM REAÇÃO REVERSÍVEL.....	54
6.2. ESTUDO DE CASO 02: DESTILAÇÃO DE ALTA PUREZA PROPENO/PROPANO ..	56
7. RESULTADOS E DISCUSSÕES.....	58
7.1 ESTUDO DE CASO 01: CSTR COM REAÇÃO REVERSÍVEL.....	58
7.2. ESTUDO DE CASO 02: DESTILAÇÃO DE ALTA PUREZA PROPENO/PROPANO ..	64
8. CONTRIBUIÇÕES E CONCLUSÕES	71
9. SUGESTÕES DE TRABALHOS FUTUROS	73
10. REFERÊNCIAS BIBLIGRÁFICAS	74
APÊNDICE - A	78

1. INTRODUÇÃO

Um dos principais objetivos da engenharia é a operação de processos no ponto ótimo ou próximo dele visando a maximização ou minimização de uma função objetivo que pode considerar restrições de operação, ambientais e/ou econômicas. A garantia da operação ótima só é possível apenas através de uma estrutura de controle bem dimensionada para manter o processo dentro das especificações dos limites de restrições e sujeito à distúrbios medidos e/ou aleatórios. O design dessa estrutura não é uma tarefa trivial porque os processos apresentam não-linearidades e complexas dependências entre as variáveis de todo processo.

Em 1980, Morari e coautores escreveram um conceito resumindo esses problemas sobre a implementação de estruturas de controle:

“... we want to find a function c of the process variables which when held constant, leads automatically to the optimal adjustments of the manipulated variables, and with it, the optimal operating conditions.”

Em tradução livre:

“... nós queremos encontrar uma função c das variáveis do processo a qual quando mantida constante, conduz automaticamente aos ajustes ótimos das variáveis manipuladas, e com isso, às condições ótimas de operação.”

Mesmo esta observação tenha sido escrita muitos anos antes, um trabalho publicado por Skogestad (2000a) propôs um procedimento sistemático para encontrar o melhor conjunto de variáveis controladas às quais, quando mantidas em constantes *setpoints*, indiretamente levariam o sistema à uma operação próxima do ótimo com uma perda aceitável. Essa metodologia foi denominada *self-optimizing control*. Após os anos 2000, um significativo número de estudos foi desenvolvido para investigar e aperfeiçoar o procedimento de *self-optimizing control*.

Recentemente, Alves et al., (2018) investigaram a vasta literatura e destacaram e concluíram que os métodos derivados e exaustivamente analisados em diversas publicações podem se tornar uma tarefa difícil quando implementado para sistemas complexos. O motivo é que para esses sistemas as avaliações para Hessianas e gradientes podem ser imprecisas, e a metodologia do *self-optimizing control* depende de uma estimativa precisa desses cálculos matemáticos. Então, os autores propuseram o uso de metamodelos para simplificar a avaliação de tais

procedimentos matemáticos com uma redução drástica de tempo. A metodologia foi implementada em uma aplicação baseada na linguagem de programação *Python* e denominada de “*Metacontrol*” (Lima et al., 2020).

Este trabalho teve por objetivo propor um procedimento sistemático, programado em linguagem *Python* no ambiente de desenvolvimento *Jupyter Notebook*, para metamodelagem, otimização e aplicação da metodologia *self-optimizing control*. O procedimento foi projetado para aplicação da metodologia SOC para processos modelados em *Python* ou no simulador de processos *Aspen Hysys* em que todo o procedimento foi automatizado. A metodologia consiste em inicialmente realizar a metamodelagem do modelo original do processo através de um design de experimentos realizado no simulador utilizando os interpoladores *Kriging*. Posteriormente, o metamodelo representativo do processo é validado para utilização nas demais etapas da metodologia. O desenvolvimento da metodologia de *self-optimizing control* é realizado para a seleção das variáveis controladas que minimizam uma função objetivo e mantém o processo operando com perda mínima mesmo na presença de distúrbios e erros de implementação. A flexibilidade de uso da ferramenta é analisada em diferentes abordagens, e avaliada para problemas da literatura.

1.1. PUBLICAÇÕES

1. Marcílio Máximo, Herbert Teixeira, Antônio Tavernard, Heleno Bispo. ***Self-optimizing control methodology using surrogate models for complex systems: A Jupyter-based application for flexible exploration and adjustment.*** Aprovado no 2023 AIChE Annual Meeting, Orlando, FL, USA.
2. Marcílio Máximo, Herbert Teixeira, Antônio Tavernard, Heleno Bispo. ***Self-Optimizing Control: Uma ferramenta usando metamodelos em aplicação Jupyter Notebook.*** Aprovado no 24º Congresso Brasileiro de Engenharia Química, Salvador, BA, Brasil.

2. OBJETIVOS

2.1. OBJETIVO GERAL

Implementar a metodologia de *Self-Optimizing Control* usando metamodelos *Kriging* em linguagem *Python* para melhoria e flexibilidade de aplicação.

2.2. OBJETIVOS ESPECÍFICOS

- Avaliar os diferentes polinômios regressores dos metamodelos na precisão dos cálculos dos gradientes e Hessianas;
- Implementar as modificações necessárias para as funções utilizadas em *Python* com o objetivo de melhorar a utilização da ferramenta pelo usuário;
- Estruturar os códigos necessário para executar ferramenta no ambiente de desenvolvimento *Jupyter Notebook*;
- Avaliar a funcionalidade da ferramenta para problemas apresentados na literatura;
- Aplicar o procedimento para implementação da metodologia de *Self-optimizing control* para processos modelado no simulador de processos *Aspen Hysys*;
- Avaliar os parâmetros que influenciam na construção de metamodelos representativos de processos modelados no simulador de processos *Aspen Hysys*;
- Disponibilizar a ferramenta com descrição detalhada e objetiva dos procedimentos visando a sua utilização para fins didáticos e de pesquisa;

3. FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo iremos apresentar as principais bases teóricas sobre os temas explorados e analisados ao longo deste trabalho. As primeiras discussões sobre o tema e os principais avanços ao longo dos últimos anos serão discutidos para que a compreensão seja completamente alcançada pelo leitor. A abordagem matemática descrita nesse capítulo segue a nomenclatura apresentada no trabalho de Jäschke et al., (2017), por ser mais didática e comumente utilizada nas publicações da área.

3.1. *PLANTWIDE CONTROL*

A primeira discussão, reconhecida pela literatura da área, sobre esse tema foi publicada no livro de Buckley (1964) no capítulo *Overall process control*. O capítulo introduz os principais problemas da teoria de controle *plantwide* e apresenta como está o desenvolvimento desse tema na perspectiva industrial. Aproximadamente uma década depois, as observações de Buckley ganharam destaque através de análises críticas feitas por Foss (1973) sobre os processos clássicos utilizados na engenharia de controle.

Foss (1973) apresenta como tese central do seu trabalho a dificuldade de aplicar as teorias de controle em processos reais, tendo em vista que esses últimos apresentam problemas que não são totalmente compreendidos. Especificamente para processos químicos, o autor destaca que o comportamento dinâmico de algumas variáveis está fortemente relacionado com inúmeras outras variáveis através de interações físico-químicas e essas correlações dinâmicas se propagam ao longo de todo processo. Muitas dessas variáveis apresentam uma dependência não-linear entre si e requerem uma atenção especial. Além disso, a definição de sistemas de controle requer a medição das variáveis do processo, mas nem todas as medições são praticáveis, em muitos casos apresentam alto custo. Dentre as variáveis medidas, estas podem estar sujeitas a erros de medição ou apresentar um tempo longo de processamento. Todos esses problemas, individualmente ou em conjunto, afetam a qualidade e a estabilidade do controle.

Nesse cenário, o trabalho de Foss (1973) destaca as suas ideias sobre a teoria de controle *plantwide*:

“The central issue to be resolved by the new theories is the determination of the control system structure.”

“Which variables should be measured, which inputs should be manipulated, and which links should be made between the two sets?”

“The gap is present indeed, but contrary to the views of many, it is the theoretician who must close it.”

Em seu trabalho Stephanopoulos (1983) também faz críticas às teorias clássicas de síntese de estrutura de controle e defende que a ideia central das configurações de controle corresponde a ações de controle *plantwide*, deixando o controle das operações unitárias para o segundo estágio de síntese. O autor observa que o problema de controle *plantwide* é multiobjetivo e que existe (naquele tempo) uma necessidade de um procedimento sistemático e organizado o qual iria identificar todos os objetivos de controle necessários.

Mesmo os estudos sobre o tema apresentarem algumas décadas de investigação, apenas a partir dos anos 2000 as pesquisas na área começaram a levar em consideração operação ótima, controladores multivariáveis e alguns outros aspectos relevantes. Diante de todas as discussões e contribuições sobre o tema, Larsson e Skogestad (2000) apresentam uma revisão detalhada da literatura sobre o tema destacando as duas principais abordagens utilizadas na literatura para implementação de uma estrutura de controle *plantwide*: a abordagem matemática e a abordagem baseada no processo.

A abordagem matemática é vastamente analisada no trabalho de Christine e Stephanopoulos (1998). Os conceitos centrais da abordagem consistem em avaliar controlabilidade, observabilidade e acessibilidade. Com base nesses conceitos, o conjunto de variáveis de entrada e medições são classificadas como viáveis ou não-viáveis. Uma desvantagem dessa abordagem é não ser quantitativa e não fornecer informação suficiente sobre a estrutura do processo que já existe.

Por outro lado, a abordagem baseada no processo foi amplamente revisada por Skogestad (2004). O autor sugere que a síntese de sistemas de controle

plantwide devem iniciar com uma análise “*top-down*” para seleção das variáveis controladas e manipuladas, e posteriormente um procedimento “*bottom-up*” para síntese da estrutura de controle. Ao final do trabalho, dez regras heurísticas foram obtidas para a síntese de controle *plantwide*.

Larsson e Skogestad (2000) propõem um novo método para síntese de estruturas de controle *plantwide* baseado nas duas abordagens, a matemática e a baseada no processo. A proposta é realizar uma análise “*top-down*” para selecionar as variáveis controladas baseadas nas ideias de *self-optimizing control* (Etapa 1). Em seguida, a estrutura econômica ótima é obtida (Etapa 2). Com as variáveis controladas e manipuladas conhecidas, uma análise “*bottom-up*” é realizada para determinar as estruturas de controle.

Mesmo com os enormes avanços ao longo das últimas décadas, o problema de *plantwide* não é perfeitamente definido por ser um problema que envolve um grande número de variáveis de decisão. Classicamente, a estrutura de controle *plantwide* envolve a seleção das variáveis controladas, das variáveis manipuladas, da configuração de controle, e do tipo de controle (Skogestad, 2012).

Recentemente, Godoy e Garcia (2017) apresentaram um trabalho de revisão sobre a estratégia de controle *plantwide* em relação às técnicas de síntese, avanços e desafios. Os autores destacam que dentre os métodos publicados nos últimos anos, a maioria deles se aplicam para processos de média e pequena escala e são adequados apenas para a seleção de estruturas de controle de unidades de processo. Para processos de grande escala, apenas algumas técnicas da literatura são aplicáveis, concluindo que ainda existem muitas oportunidades de desenvolvimento na área. Dentre as diversas oportunidades de desenvolvimento destaca-se a metodologia *self-optimizing control*

3.2. SELF-OPTIMIZING CONTROL

A metodologia *self-optimizing control* (SOC) é uma das etapas para o desenvolvimento da estrutura de controle *plantwide*. Por ser um termo consolidado na literatura da área, usaremos o termo ao longo do texto sem tradução. O conceito de *self-optimizing control* foi primeiramente proposto por Morari et. al., (1980).

“In attempting to synthesize a feedback optimizing control structure, our main objective is to translate the economic objectives into process control objectives. In other words, we want to find a function c of the process variables which when held constant, leads automatically to the optimal adjustments of the manipulated variables, and with it, the optimal operating conditions.”

Em tradução livre:

“Na tentativa de sintetizar uma estrutura de controle feedback ótima, nosso principal objetivo é traduzir os objetivos econômicos do processo em objetivos de controle. Em outras palavras, nós queremos encontrar uma função c de variáveis do processo, as quais quando mantidas constantes, conduzem automaticamente aos ajustes ótimos das variáveis manipuladas, e com isso, às condições ótimas de operação.”

O conceito do termo *self-optimizing control* foi introduzido em 1998 no trabalho de Skogestad, Halvorsen e Morud (1998), como uma estratégia para obter uma operação próxima do ótimo a partir da seleção de variáveis controladas que são simplesmente mantidas em valores constantes de *setpoint*. No entanto, o termo ganhou reconhecimento no âmbito acadêmico através do trabalho publicado por Skogestad (2000a):

“A set on controlled variables is called self-optimizing if, when it is kept at constant setpoints, the process is operated with an acceptable loss with respect to the chosen objective function (also when disturbances occur).”

Em tradução livre:

“O conjunto de variáveis controladas é chamado de *self-optimizing* se, quando são mantidas em valores constantes de *setpoint*, o processo é operado com uma perda aceitável em relação à função objetivo escolhida (também quando ocorrem distúrbios).”

É importante destacar que o termo *self-optimizing* não é uma propriedade do controlador, e sim, uma estratégia para a síntese de uma estrutura de controle, onde o objetivo principal é obter uma operação próxima do ótimo utilizando valores constantes de *setpoint* (Skogestad, 2000a; 2004). O sucesso da aplicação da

metodologia de *self-optimizing control* requer ferramentas e métodos para selecionar boas variáveis controladas (CV). A principal diferença entre essa metodologia e outros métodos de síntese de estruturas de controle, que normalmente consideram controlabilidade e performance como critério de seleção (van de Wal e Jager, 2001), é que no *self-optimizing control* a seleção das CVs é feita sistematicamente para minimizar a perda, em relação ao ponto ótimo, em relação a uma função custo econômica. Tipicamente, essa função custo está diretamente relacionada com os custos de operação, mas também a outros objetivos, como eficiência energética, ou também objetivos de controle indireto (Skogestad e Postlethwaite, 2005). Então, o procedimento de seleção das variáveis controladas é definido pela função custo, a qual é minimizada durante a operação da planta simplesmente controlando as *self-optimizing CVs* nos seus respectivos *setpoints*. Na metodologia SOC, os *setpoints* são mantidos constantes sem a necessidade de reotimização e isso é traduzido em estruturas de controle simples que mantêm o processo operando próximo da condição ótima mesmo na presença de distúrbios.

Recentemente, Jäschke e coautores (2017) realizaram uma revisão da literatura apresentando os aspectos históricos e todos os avanços realizados nos estudos da metodologia de *self-optimizing control*. Nas próximas seções apresentaremos as principais formulações, resultados e métodos desenvolvidos que irão auxiliar na compreensão acerca do tema desse trabalho de tese.

3.3. DEFININDO O PROBLEMA DE OTIMIZAÇÃO

O principal objetivo da metodologia *self-optimizing control* é dimensionar uma estrutura de controle que realize o que foi inicialmente proposto por Morari et al., (1980), ou seja, traduzir os objetivos econômicos em objetivos de controle. Essa interpretação nos revela que o controle do processo não é o objetivo final, e sim o meio para atingir a melhor performance econômica do processo para um determinado conjunto de restrições de operação e distúrbios. Matematicamente, isso pode ser formulado como um problema de otimização.

A maioria dos processos contínuos operam em estado estacionário, ou próximo deste, a maior parte do tempo, e isso faz com que a presença dos distúrbios seja desprezível para o aspecto econômico na maior parte do tempo. Assim,

podemos formular o problema de otimização como um problema de otimização em estado estacionário:

$$\min J_0(u_0, x, d) \quad (1)$$

sujeito às restrições,

$$f(u_0, x, d) = 0 \quad (2)$$

$$g(u_0, x, d) \leq 0 \quad (3)$$

Onde x representam as variáveis internas de estado, d os distúrbios do sistema, e u_0 os graus de liberdade para o estado estacionário que afetam os custos operacionais no estado estacionário, J_0 , f é a função que representa as equações do modelo, e g representa as restrições operacionais. Diversas perturbações podem afetar o sistema, por exemplo, as variações externas (i.e., variação nas condições de alimentação), variações no modelo, variações nas restrições e variações nos parâmetros da função custo (i.e., preços dos componentes). Para o problema apresentado, iremos considerar que a operação ótima é o valor de $J_0^*(d)$. Vamos considerar também que o problema de otimização é contínuo e tem um único mínimo local.

Durante a operação do processo, o custo $J_0(u_0, x, d)$ deve ser minimizado enquanto satisfaz as restrições da planta. Se todos os estados x e os distúrbios d forem conhecidos, é possível resolver o problema de otimização e aplicar os *inputs* ótimos u_0^* para a planta. Em condições ideais de operação isso irá resultar em uma operação com um custo associado $J_0^*(d)$. No entanto, na prática essa estratégia não é alcançável porque a planta não está verdadeiramente no estado estacionário, e porque o conhecimento perfeito de todos os estados e distúrbios não está disponível para o processo. Porém, as informações possíveis de um processo são provenientes de dados obtidos a partir de medições disponíveis, e aqui nós iremos considerar que temos um modelo do processo como função dessas medições:

$$y_0 = m(u_0, x, d) \quad (4)$$

Onde a função m descreve a relação entre as variáveis u_0 , x e d e os *outputs* do modelo y_0 . No entanto, os sinais y que são medidos no processo real apresentam erros de medição n^y , de tal forma que:

$$y = y_0 + n^y \quad (5)$$

3.4. IMPLEMENTAÇÃO DO PROBLEMA DE OTIMIZAÇÃO

A partir das medições y do processo, o objetivo da estrutura de controle é implementar a solução ótima do problema descrito pelas Equações 1-3 no processo real. Uma boa estrutura de controle irá garantir a operação da planta próxima do ponto ótimo econômico, mesmo na presença de variações nas condições de operação e distúrbios.

O sistema de controle de um processo químico normalmente é decomposto e organizado em uma hierarquia de atuação onde diferentes camadas de controle operam em diferentes escalas de tempo (Skogestad, 2000a). Jäschke e coautores (2017) apresentam um exemplo dessa hierarquia, apresentado na Figura 1. No topo da hierarquia está a técnica de otimização em tempo real (*real-time optimization - RTO*), que normalmente opera em uma escala de tempo de horas para calcular os *setpoints* c_s dos controladores que atuam na camada posterior em uma escala de tempo de minutos ou segundos.

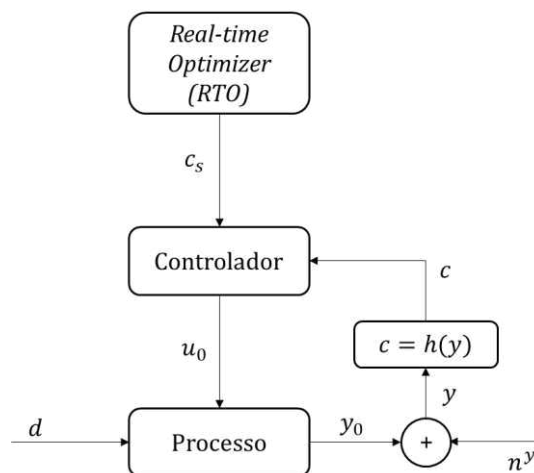


Figura 1. Hierarquia de uma estrutura de controle. (Fonte: Adaptado de Jäschke et al., 2017)

O ajuste dos *setpoints* c_s pode ser realizado pelos operadores de acordo com suas experiências operacionais do processo ou através de técnicas de otimização utilizando computadores. O controlador então ajusta os *inputs* dinamicamente para

manter as variáveis controladas (CVs) nos *setpoints* fornecidos pelo *RTO*. A variável controlada c é função das medições do processo, logo:

$$c = h(y) \tag{6}$$

A estratégia de controle influencia diretamente na escolha da variável c . A variável $c = h(y)$ pode ser função de qualquer medição do processo, mas normalmente é escolhida como uma função linear, de tal forma que $c = Hy$, onde H é matriz constante de dimensões conhecidas (Jäschke et al., 2017).

Skogestad e Postlethwaite (2005) apresentaram alguns requisitos para a escolha das variáveis controladas (CVs):

Requisito 1: a CV deve ser fácil de controlar, ou seja, os inputs u_0 devem ter um efeito significativo (ganho) na variável c .

Requisito 2: o valor ótimo de c não deve ser sensível aos distúrbios.

Requisito 3: a CV deve ser insensível aos erros de medição.

Requisito 4: para o caso de muitas CVs, as variáveis não devem possuir grande correlação.

Uma boa escolha de c que satisfaz os requisitos acima não irão requerer que a camada de *RTO* calcule os valores dos *setpoints* a cada variação das condições de operação ou distúrbios. O controle de $c = h(y)$ no seu respectivo *setpoint* indiretamente irá conduzir o processo aos *inputs* ótimos correspondentes (ou próximo dele) e a estrutura de controle é considerado como uma estrutura de *self-optimizing control*.

Ao contrário de buscar os valores ótimos dos *inputs*, na estratégia de *self-optimizing control* o objetivo é obter ou encontrar os *outputs* ótimos $c = h(y)$ da planta. Os controladores irão determinar os *inputs* correspondentes próximos do ótimo. As variações significativas de condições operacionais ou distúrbios irão requerer atualizações dos *setpoints*, no entanto, com uma boa estrutura de controle o número dessas atualizações será significativamente menor, resultando em uma boa performance sem a camada de *RTO* (Jäschke et al., 2017).

Para quantificar a performance, definimos uma perda L_c associada a uma estrutura de controle específica como a diferença entre o custo resultante dessa

estrutura de controle (representada pela escolha $c = h(y)$) e o custo resultante da verdadeira operação ótima (Halvorsen et al., 2003).

$$L_c = J_0(u_0, x, d) - J_0^*(d) \quad (7)$$

A perda L_c é utilizada para comparar estruturas de controle e avaliar se determinada estrutura é *self-optimizing*.

3.5. SELEÇÃO DAS VARIÁVEIS CONTROLADAS

A partir do trabalho publicado por Skogestad, (2000a), a metodologia de *self-optimizing control* ganhou reconhecimento na comunidade científica e vários trabalhos foram desenvolvidos na busca de métodos e estratégias para determinação das variáveis controladas que garantissem uma estrutura *self-optimizing*. A seguir apresentamos os dois principais grupos de métodos, os métodos de força bruta ou busca exaustiva e os métodos locais que determinam as *self-optimizing* CVs para o estado estacionário.

3.5.1. Método da Força Bruta ou Busca Exaustiva

Esse método consiste em calcular a perda para as possíveis combinações de variáveis controladas, e posteriormente selecionar o conjunto que apresente a menor perda. Dessa forma, é possível garantir que o resultado é ótimo para o problema de otimização. A avaliação da perda pode ser baseada na pior perda $L_{c,wc}$ ou na perda média $L_{c,av}$ associada com uma escolha particular de CVs. O pior caso é calculado por:

$$L_{c,wc} = \max L_c \quad (8)$$

E a perda média é definida como:

$$L_{c,av} = \mathbb{E}[L_c] \quad (9)$$

Onde $\mathbb{E}[\cdot]$ é o operador esperança avaliado para os conjuntos de possíveis distúrbios e erros de medição. A variável L_c representa a perda associada para o controle da variável c em um valor constante de *setpoint* para um determinado distúrbio e erro de medição.

Sabemos que na prática os conjuntos de todos os possíveis distúrbios \mathcal{D} e erros de medição \mathcal{N} geralmente têm um número infinito de pontos, podemos avaliar diferentes cenários $i = 1 \dots M$, no qual pode ser, por exemplo, distribuído uniformemente em \mathcal{D} e \mathcal{N} . Os valores de d e n^y para o i -ésimo cenário são denominados por $d^{(i)}$ e $n^{y(i)}$, respectivamente, e o valor ótimo da função custo correspondente $J_0^*(d^{(i)})$ é obtido pela solução do problema apresentado nas Equações 1-3.

Dado um conjunto de candidatas a CV $c_{(j)}$ e seus valores de *setpoints* $c_{s(j)}$, para cada cenário $i = 1 \dots M$ o custo $J_0^{(i)}(j)$ obtido pelo controle de $c_{(j)}$ em seus *setpoints* é determinado pela solução do problema de otimização:

$$\min J_0(u_0^{(i)}, x^{(i)}, d^{(i)}) \quad (10)$$

sujeito a,

$$f(u_0^{(i)}, x^{(i)}, d^{(i)}) = 0 \quad (11)$$

$$g(u_0^{(i)}, x^{(i)}, d^{(i)}) \leq 0 \quad (12)$$

$$c_{(j)}(u_0^{(i)}, x^{(i)}, d^{(i)}, n^{y(i)}) - c_{s(j)} = 0 \quad (13)$$

Tipicamente, os *setpoints* são escolhidos como os valores ótimos nominais do processo que são obtidos através da otimização do sistema para $d = d_{nom}$ e $n^y = 0$. Todos os conjuntos candidatos à variáveis controladas $c_{(j)}$, $j = 1 \dots n_j$ são ranqueados a partir das Equações (8) e (9). Por exemplo,

$$L_{c_{(j)},wc} = \max [J_0^{(i)}(j) - J_0^*(d^{(i)})] \quad (14)$$

$$L_{c_{(j)},av} = \mathbb{E} [J_0^{(i)}(j) - J_0^*(d^{(i)})] \quad (15)$$

O método da força bruta requer a solução de muitos problemas de otimização. O problema apresentado nas Equações 10 – 13 precisa ser resolvido para cada candidata a CV $j = 1 \dots n_j$ para todos os $i = 1 \dots M$ cenários. Quando selecionamos n_{u_0} medições como CVs de um total de n_y medições, existem um número de combinações possíveis de estruturas de controle que podem ser escolhidas,

$$C_{(n_y)}^{(n_{u_0})} = \frac{n_y!}{(n_y - n_{u_0})! n_{u_0}!} \quad (16)$$

Logo, o número de possíveis estruturas de controle aumenta significativamente com o número de medições n_y . Essa situação é comum em processos reais, logo, esse procedimento se torna inviável. No entanto, o método tem como vantagem que a obtenção do conjunto ótimo de variáveis controladas será inevitavelmente determinado, pois todas as combinações possíveis serão analisadas e quantificadas. O método da força bruta foi aplicado para uma planta de larga escala, o *The Tennessee Eastman Process* (Larsson et al., 2001).

3.5.2. Método Exato Local

Os métodos locais foram desenvolvidos para limitar o número de CVs que serão analisadas e para eliminar as CVs que não apresentam boa performance na escolha inicial da estrutura de controle. A principal vantagem desses métodos é que a candidata à variável controlada deve ter uma boa performance de controle localmente em torno do ponto ótimo operacional, ou seja, onde o processo irá atuar a maior parte do tempo de operação. Apenas as CVs que apresentam melhores performances próximas das condições nominais são então examinadas e testadas para toda região de operação.

3.5.2.1. Aproximação local da função custo

A partir do problema apresentado nas Equações (1-3), em estado estacionário, as equações do modelo $f(u_0, x, d) = 0$ são usadas para formalmente eliminar os estados x do problema de otimização. Essa aproximação reduz o problema a,

$$\min J_0^r(u_0, d) \quad (17)$$

sujeito à restrição,

$$g(u_0, d) \leq 0 \quad (18)$$

De acordo com Skogestad, (2000a), as variáveis ativas precisam ser consideradas que não irão variar durante a operação, e a primeira etapa do problema é controlar as variáveis ativas em seus valores ótimos. Assim, as variáveis

ativas podem ser formalmente eliminadas do problema pela utilização dos graus de liberdade correspondentes. O problema então resulta em,

$$\min J(u, d) \quad (19)$$

Onde $J(u, d)$ é a função custo do problema reduzido, e u representa os graus de liberdade disponíveis após o controle das variáveis ativas. Esses graus de liberdade serão usados para controlar as *self-optimizing* CVs. Aqui consideramos que o valor ótimo da função custo é $J^*(d)$, e podemos observar que, se os distúrbios d não provocam mudanças nas variáveis ativas, temos que

$$J^*(d) = J_0^*(d) \quad (20)$$

Para determinar boas CVs candidatas a variáveis *self-optimizing*, o problema de otimização não linear da Equação (19) é aproximado localmente por uma função quadrática em torno do ponto ótimo nominal. Introduzindo as variáveis desvio,

$$\Delta u = u - u_{nom} \quad (21)$$

$$\Delta d = d - d_{nom} \quad (22)$$

Onde u_{nom} é o valor do *input* no ótimo nominal correspondente ao distúrbio nominal d_{nom} . Uma expansão de séries de Taylor em torno do ponto ótimo nominal resulta em,

$$J(\Delta u, \Delta d) \approx J_{nom} [J_u \ J_d] \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} + \frac{1}{2} [\Delta u^T \ \Delta d^T] \begin{bmatrix} J_{uu} & J_{ud} \\ J_{du} & J_{dd} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} \quad (23)$$

Onde, $J_u = \frac{\partial J}{\partial u}$ e $J_d = \frac{\partial J}{\partial d}$ são as primeiras derivadas da função custo em relação a u e d , respectivamente, e $J_{uu} = \frac{\partial^2 J}{\partial u^2}$, $J_{ud} = J_{du} = \frac{\partial^2 J}{\partial u \partial d}$ e $J_{dd} = \frac{\partial^2 J}{\partial d^2}$ são as segundas derivadas da função custo, avaliadas em $u = u_{nom}$ e $d = d_{nom}$.

Como estamos aproximando a função custo em torno do ponto ótimo nominal, temos que $J_u = 0$. Diferenciando a Equação (23) em relação a Δu e igualando a equação a zero teremos a condição ótima que deve ser satisfeita para que o problema permaneça no ponto ótimo,

$$\frac{dJ}{du} \approx J_u + J_{uu}\Delta u + J_{ud}\Delta d = [J_{uu} \ J_{ud}] \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} = 0 \quad (24)$$

Observe que J_u , J_{uu} e J_{ud} são avaliadas no ponto ótimo nominal, enquanto $\frac{dJ}{du}$ é o valor do gradiente em um ponto próximo para variações de Δu e Δd . Resolvendo essa equação para o *input* ótimo $\Delta u^*(d)$, teremos,

$$\Delta u^*(d) = -J_{uu}^{-1} J_{ud} \Delta d \quad (25)$$

Usando a função quadrática do custo da Equação (23) e o *input* ótimo da Equação (25), podemos observar que a perda em relação ao ponto ótimo é expressa como (Alstad, 2005),

$$L = J(u, d) - J^*(d) = \frac{1}{2} (\Delta u - \Delta u^*(d)) J_{uu} (\Delta u - \Delta u^*(d)) \quad (26)$$

Ou alternativamente,

$$L = \frac{1}{2} \|z\|_2^2 \quad (27)$$

Onde, $\|\cdot\|_2$ representa a norma Euclidiana com z definido como,

$$z = J_{uu}^{-1/2} (\Delta u - \Delta u^*(d)) \quad (28)$$

Essa expressão da perda apresentada na Equação (27) é a base para os métodos locais apresentados nessa seção.

Podemos analisar ainda que a combinação da Equação (25) na Equação (28), obtemos,

$$z = J_{uu}^{-1/2} [J_{uu} \ J_{ud}] \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} \quad (29)$$

Comparando a Equação (29) com a expressão do gradiente na Equação (24) podemos observar que a perda local da Equação (27) pode ser escrita como o quadrado ponderado do comprimento Euclidiano do valor do gradiente:

$$L = \frac{1}{2} \left\| J_{uu}^{-1/2} [J_{uu} \ J_{ud}] \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} \right\|_2^2 = \frac{1}{2} \left\| (J_{uu}^{-1/2}) \frac{dJ}{du} \right\|_2^2 \quad (30)$$

Onde a expressão do gradiente $\frac{dJ}{du}$ é o gradiente atual da planta em operação, não no ponto ótimo. Podemos observar que quando o gradiente é zero, condição ótima necessária, então a perda local será zero.

3.5.2.2. Aproximação local da combinação linear das medições

O modelo para as medições pode ser linearizado em torno do ponto ótimo nominal a partir da Equação (5), logo,

$$\Delta y = G^y \Delta u + G_d^y \Delta d + n^y \quad (31)$$

Onde, $\Delta y = y - y_{nom}$, e $G^y = \frac{\partial y}{\partial u}$, e $G_d^y = \frac{\partial y}{\partial d}$ são as matrizes Jacobiana de tamanhos conhecidos, que são avaliadas em torno do ponto ótimo nominal, o qual representam o ganho dos *inputs* e distúrbios, respectivamente, em relação aos *outputs*. Seleccionado CVs como uma combinação linear das medições y , teremos

$$\Delta c = H \Delta y \quad (32)$$

Onde, H é a matriz de combinação das medições. Combinando as Equações (31) e (32), teremos,

$$\Delta c = H G^y \Delta u + H G_d^y \Delta d + H n^y \quad (33)$$

Geralmente, os elementos da matriz H pode assumir valores aleatórios, de tal forma que o $rank(HG^y) = n_u$. Essa condição é necessária para obter um conjunto de CVs linearmente independentes que satisfaz completamente o sistema. Se um único conjunto de variáveis são seleccionadas como CVs, então cada linha da matriz H terá exatamente um número "1", e os demais elementos da linha serão zero. Nesse caso, H tem a propriedade,

$$H H^y = I_{n_u} \quad (34)$$

Onde, I_{n_u} é a matriz identidade de dimensões $n_u \times n_u$.

3.5.2.3. Método Exato Local

A análise da perda correspondente para uma estrutura de controle específica, devemos substituir o *input* que é gerado para controlar $\Delta c = H \Delta y$ em zero, com a expressão para o *input* ótimo da Equação (25) na expressão da perda da Equação (26). Para um dado distúrbio Δd e uma dada estrutura de controle representada por

H , o *input* gerado pelos controladores pode ser encontrado resolvendo a Equação (33) para Δu ,

$$\Delta u = (HG^y)^{-1}(\Delta c - HG_d^y \Delta d - Hn^y) \quad (35)$$

Como $\Delta c = 0$,

$$\Delta u = -(HG^y)^{-1}H(G_d^y \Delta d - n^y) \quad (36)$$

Inserindo o *input* obtido pela Equação (36) e o *input* ótimo da Equação (25) nas expressões da perda das Equações (26) e (27), a variável da perda z para um dado distúrbios e uma dada matriz H de combinações das medições se torna,

$$z = -J_{uu}^{1/2}(HG^y)^{-1}H[(G_d^y - G^y J_{uu}^{-1} J_{ud})\Delta d - n^y] \quad (37)$$

Essa expressão pode ser simplificada para,

$$z = -J_{uu}^{1/2}(HG^y)^{-1}H[F\Delta d - n^y] \quad (38)$$

Onde usamos a matriz de sensibilidade ótima,

$$F = G_d^y - G^y J_{uu}^{-1} J_{ud} \quad (39)$$

Para uma dada matriz H combinações de medições, um distúrbio Δd e um erro de medição n^y , a perda é calculada como $L = \frac{1}{2} \|z\|_2^2$.

A matriz F representa a sensibilidade dos valores ótimos das medições em relação aos distúrbios.

$$F = \frac{\partial y^*}{\partial d} \quad (40)$$

Podemos obter a comprovação da Equação (40) substituindo o *input* ótimo Δu^* da Equação (25) na Equação (31) das medições Δy para obtermos a variação ótima das medições Δy^* para um distúrbio Δd ,

$$\begin{aligned} \Delta y^* &= y^* - y_{nom} = -G^y J_{uu}^{-1} J_{ud} \Delta d + G_d^y \Delta d + n^y \\ \Delta y^* &= (-G^y J_{uu}^{-1} J_{ud} + G_d^y) \Delta d + n^y \\ \Delta y^* &= F \Delta d + n^y \end{aligned} \quad (41)$$

Note que, $\Delta d = d - d_{nom}$, e pela diferenciação da Equação (41) em relação a d , nós obtemos $F = \frac{\partial y^*}{\partial d}$.

Para avaliar a performance de um dado conjunto de CVs, $\Delta c = H\Delta y$ para uma variação de distúrbios, as matrizes diagonais de magnitude são definidas para os distúrbios e os erros de medição, W_d e W_n , respectivamente, como,

$$\Delta d = W_d d'; n^y = W_n n' \quad (42)$$

Onde, d' e n' representam os valores escalares dos distúrbios e dos erros de medição, respectivamente. A representação escalar dessas variáveis permite fazer observações sobre a perda para um dado conjunto de distúrbios e erros de medição definidos em W_d e W_n .

Usando as matrizes de magnitude, a perda da Equação (27) para valores específicos de d' e n' é,

$$L = \frac{1}{2} \left\| J_{uu}^{1/2} (HG^y)^{-1} H Y \begin{bmatrix} d' \\ n' \end{bmatrix} \right\|_2^2 \quad (43)$$

Onde,

$$Y = [F W_d \quad W_n] \quad (44)$$

Para uma notação simplificada, é introduzida a matriz da perda M como,

$$M = J_{uu}^{1/2} (HG^y)^{-1} H Y \quad (45)$$

Então a Equação (43) pode ser escrita como,

$$L = \frac{1}{2} \left\| M \begin{bmatrix} d' \\ n' \end{bmatrix} \right\|_2^2 \quad (46)$$

Dependendo de como os distúrbios e os erros de medição sejam considerados e assumidos, podemos ter diferentes casos para a pior perda e a perda média. A seguir, serão apresentadas as abordagens comumente utilizadas nas análises de *self-optimizing* para determinação das melhores CVs. As abordagens se diferenciam apenas na forma que os vetores de distúrbios e erros de medição são escalados.

Norma Euclidiana com restrição para os distúrbios e erros de medição

(Halvorsen et al., 2003)

Quando consideramos que os distúrbios e os erros de medição são independentes e estão uniformemente distribuídos para o conjunto,

$$\mathcal{DN}_2 = \{(d', n') \mid \|[d', n']^T\|_2 \leq 1\} \quad (47)$$

A pior perda é derivada baseada na Eq. (46),

$$\begin{aligned} L_{wc} &= \max_{\|[d', n']\|_2 \leq 1} \frac{1}{2} \|[M \begin{bmatrix} d' \\ n' \end{bmatrix}]\|_2^2 = \frac{1}{2} \|M\|_2^2 \\ L_{wc} &= \frac{1}{2} \bar{\sigma}^2(M) \end{aligned} \quad (48)$$

Onde, $\bar{\sigma}(\cdot)$ representa o maior valor singular da matriz.

Norma do Máximo com restrição para os distúrbios e erros de medição (Kariwala et al., 2008)

Se os distúrbios e os erros de medição d' e n' são considerados como independentes e estão uniformemente distribuídos para o conjunto,

$$\mathcal{DN}_\infty = \{(d', n') \mid \|[d', n']^T\|_\infty \leq 1\} \quad (49)$$

Onde, $\|\cdot\|_\infty$ é a norma do máximo, a perda média (calculada pelo operador esperança $\mathbb{E}[\cdot]$) é dada por,

$$\begin{aligned} L_{av} &= \mathbb{E} \left[\frac{1}{2} \|[M \begin{bmatrix} d' \\ n' \end{bmatrix}]\|_2^2 \right] \\ L_{av} &= \frac{1}{6} \|M\|_F^2 \end{aligned} \quad (50)$$

Onde, $\|\cdot\|_F$ é a norma de Frobenius.

Distúrbios e erros de medição normalmente distribuídos (Kariwala et al., 2008)

Considerando que d' e n' são normalmente distribuídos com média zero e variância unitária,

$$\mathcal{DN}_{\mathcal{N}} = \{d' \sim \mathcal{N}(0,1) \quad n' \sim \mathcal{N}(0,1)\} \quad (51)$$

A pior perda será sem restrição $L_{worst} = \infty$, quando d' e n' se tornam arbitrariamente grandes. A perda média, no entanto, será

$$L_{av,\mathcal{N}} = \mathbb{E} \left[\frac{1}{2} M \begin{bmatrix} d' \\ n' \end{bmatrix} \right]$$

$$L_{av} = \frac{1}{2} \|M\|_F^2 \quad (52)$$

Mesmo com as diferentes expressões para o calcula das perdas, Jäschke e coautores (2017) destacam que esses resultados não apresentam sentido prático, tendo em vista que o problema foi formulado de tal forma que o valor da perda é escalado com o número de medições n_y . Isso resulta em resultados incoerentes porque o valor da perda pode variar arbitrariamente, por exemplo, adicionando medições (aumentando n_y) enquanto adiciona colunas correspondentes de zeros na matriz H .

3.5.2.4. Determinando a combinação ótima das medições

A partir das equações e desenvolvimentos apresentados nas seções anteriores, é possível avaliar a perda média e a pior perda correspondente a uma estrutura de controle representada por H . Esse desenvolvimento matemático nos fornece ferramentas para encontrar uma estrutura de controle dentre um conjunto de possíveis CVs que nos fornece uma perda mínima. Por exemplo, para determinar a combinação de medições $\Delta c = H\Delta y$ que minimiza a perda média para distúrbios e erros de medição normalmente distribuídos, podemos escrever,

$$\min_H L = \frac{1}{2} \|J_{uu}^{-1/2} (HG^y)^{-1} HY\|_F^2 \quad (53)$$

Desde que HG^y seja invertível.

A princípio, qualquer expressão para a perda das Equações (48), (50) e (52) pode ser usada na função custo da Equação (53) para determinar a matriz de combinações. Em geral, as expressões das perdas para problemas de otimização não-convexos, que resulta em múltiplos ótimos, são difíceis de resolver. No entanto, em casos em que H pode assumir valores arbitrários, ou quando a estrutura de H é preservada na multiplicação à esquerda com qualquer matriz não singular, o

problema não convexo pode ser simplificado em um problema convexo, de tal forma que a combinação de medições ótimas H é facilmente determinada.

A seguir alguns métodos aplicados para determinação da combinação das medições são apresentados.

Método do Espaço Nulo (Null-space method) (Alstad e Skogestad, 2007)

No método do espaço nulo para determinação da matriz H , considera-se que os erros de medição podem ser desprezados ($W_n = 0$), então é preciso apenas compensar as variações causadas pelos distúrbios d . Além disso, considera-se casos em que temos $n_y \geq n_u + n_d$ variáveis independentes. Ou seja, o número de medições é maior que o número de graus de liberdade (*inputs*) mais os distúrbios. Com essas considerações, é possível determinar a matriz H que resulta em uma perda zero, simplesmente selecionando H no espaço nulo à esquerda da matriz F , de tal modo que,

$$HF = 0 \quad (54)$$

Para mostrar essa equação, consideramos o caso com pior perda $L_{wc} = \frac{1}{2} \bar{\sigma} (J_{uu}^{-1/2} (HG^y)^{-1} W_d Y)^2$. Como não consideramos erro de medição $W_n = 0$, de tal modo que $Y = [FW_d \ 0]$, e a pior perda resultante pode ser simplificada para $L_{wc} = \frac{1}{2} \bar{\sigma} (J_{uu}^{-1/2} (HG^y)^{-1} HFW_d)^2$. Selecionando H para garantir que $HF = 0$ resulta em uma pior perda e perda média $L_{wc} = L_{av} = 0$, desde que HG^y seja não singular.

A aplicação do método do espaço nulo não resulta necessariamente em uma operação ótima real porque os erros de medição não são considerados em seu desenvolvimento, conforme já explicitado nas considerações sobre o método. Além disso, a necessidade de que se tenha mais medições que a soma dos *inputs* e distúrbios resulta em complexas estruturas de controle envolvendo muitas variáveis.

Método do Espaço Nulo Estendido (Extended Null-space method) (Alstad et al., 2009)

Como discutido anteriormente, o método do espaço nulo apresenta algumas aproximações. Alstad et al., (2009) derivou uma modificação do método do espaço nulo para considerar os erros de medição.

Primeiro utilizamos a matriz escalar W_n para definir a matriz ajustada das medições y' , de tal forma que,

$$y' = W_n^{-1} \Delta y \quad (55)$$

Na presença de erros de medição n_y , a expressão para a matriz ajustada das medições é calculada usando a Equação (31) e $\tilde{G}^y = [G^y \quad G_d^u]$ como,

$$y' = W_n^{-1} \Delta y$$

$$y' = W_n^{-1} \tilde{G}^y \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} + W_n^{-1} n^y \quad (56)$$

Após rearranjarmos,

$$W_n^{-1} n^y = -W_n^{-1} \Delta y + W_n^{-1} \tilde{G}^y \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} \quad (57)$$

A solução dos mínimos quadrados $[\Delta u \quad \Delta d]^T$ para Equação (57), na qual minimiza a soma dos quadrados para os erros de medição ponderados $n^{yT} W_n^{-1} W_n^{-1} n^y$ é dada por,

$$\begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} = (W_n^{-1} \tilde{G}^y)^T W_n^{-1} \Delta y \quad (58)$$

Inserindo essa expressão da solução dos mínimos quadrados na expressão para o gradiente na Equação (24), nós obtemos,

$$c = [J_{uu} \quad J_{ud}] (W_n^{-1} \tilde{G}^y)^T W_n^{-1} \Delta y \quad (59)$$

E a matriz H correspondente é,

$$H = [J_{uu} \quad J_{ud}] (W_n^{-1} \tilde{G}^y)^T W_n^{-1} \quad (60)$$

Exceto para um fator escalar constante que não afeta a perda, essa é a mesma expressão originalmente derivada em Alstad et al., (2009). Selecionando H a partir da Equação (60), os vetores bases no espaço nulo à esquerda de F são os escolhidos para minimizar os efeitos dos erros de medição (Jäschke et al., 2017).

É importante destacar que o método do espaço nulo estendido possui duas importantes desvantagens. Primeiro, o método apenas funciona quando nós temos mais medições disponíveis que a soma dos números de *inputs* e distúrbios, assim, um grande número de medições é necessário. Segundo, o método do espaço nulo estendido geralmente não irá fornecer um *trade-off* ótimo entre a rejeição dos erros de medição e os distúrbios. Por exemplo, em alguns casos pode ser mais interessante rejeitar um distúrbio completamente do que compensar os efeitos dos erros de medição (Jäschke et al., 2017).

3.5.2.5. Selecionando os subconjuntos das medições

Em muitos casos práticos, usar uma combinação de todas as medições disponíveis não é algo mandatário ou necessário. Normalmente, o controle de um subconjunto das medições resulta em uma performance similar quando se usa combinações de todas as medições possíveis (Kariwala, 2007; Kariwala et al., 2008). O resultado é uma estrutura de controle simples que normalmente apresenta um aumento não significativo da perda.

A seleção do melhor subconjunto de n variáveis dentre as n_y medições disponíveis nos leva a um problema de otimização combinatória, onde a perda L tem que ser avaliada individualmente para cada variável controlada. O número de possíveis combinações pode ser tornar muito grande e a seleção de n combinações de CVs a partir de um conjunto de n_y medições nos leva a $C_{n_y}^n = \binom{n_y}{n} = \frac{n_y!}{(n_y-n)!n!}$ possíveis estruturas.

Na literatura, o problema de seleção dos melhores subconjuntos das medições tem sido abordado através do algoritmo do *Branch and Bound* apresentado nos trabalhos desenvolvidos em Cao e Kariwala, (2008); Kariwala e Cao, (2009, 2010).

Algoritmo do Branch and Bound

Esse método explora a monotonicidade do valor da perda ótima como uma função do número de variáveis. Isso significa que quando umas das medições é removida dentre as possíveis CVs, a perda ótima não pode diminuir. A seguir é apresentada a ideia geral para minimização da perda média para distúrbios e erros de medição normalmente distribuídos. A descrição detalhada e o tratamento da pior perda é analisada nos trabalhos de Cao e Kariwala, (2008); Kariwala e Cao, (2009, 2010).

Considerando X para representar um conjunto de medições selecionadas e $|X|$ a sua cardinalidade (representação matemática para indicar o número de elementos de um conjunto). Considerando também que G_X^y representa o ganho da matriz correspondente para as medições selecionadas, ou seja, as linhas de G^y correspondem aos índices em X . Analogamente, Y_X representa a matriz que consiste das linhas em Y correspondentes aos índices em X , e H_X^* representa a matriz H ótima correspondente ao subconjunto selecionado das medições.

Então, para um dado conjunto de distúrbios e erros de medição normalmente distribuídos, a perda média ótima é uma função do conjunto de índices X ,

$$L_{av}(X) = \frac{1}{2} \left\| J_{uu}^{-1/2} (H_X^* G_X^y)^{-1} H_X^* Y_X \right\|_F^2 \quad (61)$$

Essa perda média ótima está diminuindo monotonicamente com o aumento do número de medições, porque a adição de uma medição para a combinação pode apenas melhorar a performance ótima.

Usando as propriedades de monotonicidade, um algoritmo eficiente pode ser desenvolvido para a seleção do melhor subconjunto das medições. Agora considerando X_{all} o conjunto de índices de todas as medições disponíveis, e X_n qualquer conjunto correspondente a seleção de n medições. O conjunto de índices ótimos para a melhor CV contendo n variáveis é,

$$X_n^* = \min_{X_n \subset X_{all}} L_{av}(X_n) \quad (62)$$

A propriedade de monotonicidade implica que se um conjunto de índices X_n está contido em outro conjunto de índices denominado por X_m ($X_n \subset X_m$) então a

perda ótima do superconjunto deve ser menor ou igual a perda ótima do subconjunto,

$$L(X_m) \leq L(X_n) \quad (63)$$

Assumindo que B é o limite superior para a perda mínima para o caso em que n variáveis são selecionadas, de tal forma que,

$$B \geq L(X_n) \quad (64)$$

Então esse limite superior pode ser usado para excluir conjuntos de medições. No caso em que $X_n \subset X_m$, sempre será possível observar,

$$L(X_m) \geq B \quad (65)$$

de tal forma que,

$$L(X_m) \geq L(X_n) \quad (66)$$

E todos os subconjuntos incluídos em X_m podem ser excluídos da avaliação, porque esses conjuntos não irão resultar em uma perda ótima menor. A Equação (65) é considerada uma condição de eliminação (*pruning condition*). Essa condição pode ser usada em algoritmos para eliminação sistemática de subconjuntos de medições subótimas para consideração.

Métodos mais sofisticados do algoritmo *branch and bound* foram desenvolvidos utilizando critérios de eliminação bidirecional, o que torna o método mais eficiente (Kariwala e Cao, 2009; 2010). O algoritmo desenvolvido em Matlab está disponível para *download* gratuito na internet¹.

3.6. APLICAÇÃO DA METODOLOGIA DE *SELF-OPTIMIZING CONTROL*

Muitas das análises de processos utilizando a metodologia de *self-optimizing control* são aplicadas para modelos de processos químicos desenvolvidos em diferentes linguagens de programação, por exemplo, modelos desenvolvidos em *Matlab* (Jäschke et al., 2011; Araújo et al., 2013; Graciando et al., 2015).

No entanto, esses modelos também podem ser desenvolvidos em *softwares* de simulação de processos, dentre os vários disponíveis no mercado, por exemplo, *Aspen Plus*, *Aspen Hysys*, *Aveva PRO/II*, *Aveva Process Simulation*, *UniSim*, *Chemcad*. A

¹ O download do código pode ser realizado no endereço eletrônico:
https://www.mathworks.com/matlabcentral/fileexchange/25870-bidirectional-branch-and-bound-for-average-loss-minimization?s_tid=srchtitle

aplicação da metodologia SOC para modelos desenvolvidos em *softwares* de simulação requer o desenvolvimento da automação da aquisição de dados e manipulação parâmetros dos simuladores, e nesse cenário algumas dificuldades podem surgir. Dentre essas dificuldades, podemos citar o tempo excessivo e o esforço computacional necessários para execução de estudos de casos complexos e o número de simulações necessárias para implementação da metodologia. Além disso, a metodologia SOC é baseada nas estimativas de gradientes e Hessianas do processo, e matematicamente o cálculo dessas derivadas é limitado pela tolerância dos simuladores.

Nesse contexto, alguns trabalhos da literatura avaliaram a modelagem de processos utilizando simuladores comerciais para aplicação da metodologia SOC. O trabalho desenvolvido por Araujo et al., (2007) aplicou a metodologia para o processo de uma planta de hidrodealquilação de tolueno (*HDA*) utilizando o simulador comercial *Aspen Plus* com o auxílio do *Microsoft Excel* para aquisição de dados usando comunicação *OLE (Object Linking and Embedding)*. O cálculo dos gradientes e Hessianas foram realizados no *Matlab* utilizando aproximação por diferenças finitas de primeira ordem, no entanto, essa aproximação apresenta alta sensibilidade a ruídos nos dados. Por outro lado, visando evitar o problema da sensibilidade aos ruídos, Silva et al., (2017) utilizou o método *Akima* com *spline* cúbica e *spline* bicúbica para a estimativa dos gradientes e Hessianas, respectivamente. No entanto, os autores destacam que esse método tem como desvantagem que a presença de um ponto fora da região de interesse prejudica a avaliação em alguns casos. Este trabalho utilizou o simulador *Aveva PRO/II* para avaliar a metodologia SOC em uma coluna de destilação deetanizadora.

Na mesma sequência de estudos abordando as dificuldades na implementação da metodologia SOC, em especial para modelos de processos desenvolvidos em simuladores de processo, Alves et al., (2018) desenvolveram um procedimento sistemático que utiliza metamodelos usando interpoladores *Kriging* para representar o processo e expressões explícitas e analíticas para o cálculo de derivadas. No estudo, o modelo desenvolvido nos simuladores é utilizado para o treinamento de metamodelos que posteriormente são utilizados para os procedimentos de otimização e estimativa de gradientes e Hessianas. Os resultados

apresentados foram validados com problemas da literatura e o procedimento se mostrou eficiente para a determinação dos conjuntos de variáveis controladas.

Nas próximas seções apresentaremos o desenvolvimento matemático para os metamodelos utilizando interpoladores *Kriging*, e suas principais aplicações e contribuições significativas para a implementação da metodologia de *self-optimizing control*.

3.7. METAMODELOS: KRIGING

Existem diversas técnicas de aprendizado de máquina e identificação de modelos. O *Kriging* é um processo Gaussiano de regressão que trata o problema de aproximação a partir de ponto de vista probabilístico. É considerado um método de aprendizado de máquina que permite a representação de sistemas a partir de processos gaussianos combinados com modelos de regressão (Williams e Rasmussen, 2006).

O *Kriging* pode ser considerado como uma técnica de interpolação que leva em consideração a correlação espacial entre os pontos de dados próximos uns aos outros. Ele é capaz de capturar a variabilidade espacial e as tendências do sistema modelado. O interpolador *Kriging* foi primeiramente proposto por (Krige, 1951) e posteriormente foi formulado na perspectiva de engenharia no trabalho desenvolvido por Sacks et al., (1989), que aplicou o método para aproximação de experimentos realizados em computadores. A implementação da metodologia foi desenvolvida por Lophaven et al., (2002) em *Matlab* a qual foi nomeada *DACE Toolbox*.

A metodologia consiste em um conjunto de m experimentos $S = [s_1 \dots s_m]^T$ com $s_i \in \mathbb{R}^n$ com as respostas $Y = [y_1 \dots y_m]^T$ com $y_i \in \mathbb{R}^q$. Um modelo matemático $\hat{y}(x)$ que representa a resposta de uma função determinística $y_x \in \mathbb{R}^q$ para uma entrada n dimensional $x \in \mathcal{D} \subseteq \mathbb{R}^n$, como a realização de um modelo de regressão \mathcal{F} e uma função aleatória de um processo estocástico z é

$$\hat{y}_l(x) = \mathcal{F}(\beta_{:,l}, x) + z_l(x), \quad l = 1, \dots, q \quad (67)$$

Usa-se um modelo de regressão o qual é uma combinação linear de p funções $f_j: \mathbb{R}^q \mapsto \mathbb{R}$,

$$\begin{aligned}\mathcal{F}(\beta_{:,l}, x) &= \beta_{1,l}f_1(x) + \dots + \beta_{p,l}f_p(x) \\ \mathcal{F}(\beta_{:,l}, x) &= \beta_{:,l}[f_1(x) \dots f_p(x)] \\ \mathcal{F}(\beta_{:,l}, x) &= \beta_{:,l}f(x)^T\end{aligned}\tag{68}$$

Tipicamente, $f(x)$ é escolhido como polinômios com ordem variando de zero a dois. Os coeficientes $\beta_{:,l}$ são parâmetros de regressão e z é considerada como uma função de média zero e covariância entre dois pontos w e x :

$$\text{Cov}[z_l(w), z_l(x)] = \sigma_l^2 \mathcal{R}(\theta_l, w, x), \quad l = 1, \dots, q\tag{69}$$

onde σ_l^2 é a variância do processo para o componente l th da resposta e $\mathcal{R}(\theta_l, w, x)$ é o modelo de correlação espacial como parâmetros θ . No entanto, é importante mencionar que o valor verdadeiro para a resposta é escrito como

$$y_l(x) = \mathcal{F}(\beta_{:,l}, x) + \alpha(\beta_{:,l}, x)\tag{70}$$

onde α é o erro de aproximação.

Para os interpoladores *Kriging*, o modelo de correlação é a seguinte equação matemática:

$$\mathcal{R}(\theta_l, w, x) = \exp(-\sum_{i=1}^m \theta_l (w - x_i)^{p_l})\tag{71}$$

$$\theta_l \geq 0, p_l \in [0, 2]$$

Para a avaliação dos gradientes e Hessianas, derivadas por Lophaven et al., (2002) e Alves et al., (2018), respectivamente, podem ser descritas pelas equações a seguir:

$$\hat{y}'(x) = J_f(x)^T \beta^* + J_r(x)^T \gamma^*\tag{72}$$

$$\hat{y}''(x) = H_f(x)^T \beta^* + H_r(x)^T \gamma^*\tag{73}$$

Onde, J_f e J_r são as Jacobianas para os modelos de regressão e correlação respectivamente; H_f e H_r são as Hessianas para os modelos de regressão e

correlação respectivamente; β^* são os parâmetros de regressão estimados; γ^* são as constantes estimadas dos parâmetros de correlação.

3.8. OTIMIZAÇÃO: *INTERIOR POINT OPTIMIZER (IPOPT)*

O método de otimização denominado de *IPOPT* é um algoritmo de otimização não-linear desenvolvido no trabalho de Wächter (2002). O algoritmo pode ser utilizado para problemas de otimização de larga escala para resolver problemas de otimização linear na forma,

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f(x) \\ \text{para, } & g^L \leq g(x) \leq g^U \\ & x^L \leq x \leq x^U \end{aligned} \quad (74)$$

Onde, $x \in \mathbb{R}^n$ são as variáveis de otimização com os limites inferiores e superiores, $x^L \in (\mathbb{R} \cup \{-\infty\})^n$ e $x^U \in (\mathbb{R} \cup \{+\infty\})^n$, $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ é a função objetivo, e $g(x): \mathbb{R}^n \rightarrow \mathbb{R}^m$ são as restrições. As funções $f(x)$ e $g(x)$ podem ser lineares ou não-lineares e convexas ou não-convexas, mas deve ser suficientemente suave (no mínimo uma, idealmente duas vezes diferenciável). As restrições, $g(x)$, têm limites inferiores e superiores, $g^L \in (\mathbb{R} \cup \{-\infty\})^m$ e $g^U \in (\mathbb{R} \cup \{+\infty\})^m$.

O *IPOPT* implementa e segue uma abordagem de ponto interior, evitando assim a complexidade combinatória de identificação das restrições ativas. A ênfase é colocada em flexibilidade no cálculo das direções de busca, o que permite a adaptação do método para aplicações individuais e é obrigatório para a solução de grandes problemas Wächter (2002). Os detalhes matemáticos do método estão além do escopo deste trabalho, mas podem ser encontrados em detalhes na literatura (Wächter, 2002; Wächter e Biegler, 2005a; Wächter e Biegler, 2005b; Wächter e Biegler, 2006; Nocedal et al., 2009).

O *IPOPT* é parte de uma iniciativa da fundação *Computational Infrastructure for Operations Research (COIN-OR Initiative)*. O projeto foi implementado em C++ e está disponível como um código *open-source* e as informações detalhadas estão disponíveis na página oficial². Um tutorial está disponível para auxílio na utilização do algoritmo com exemplos de aplicações (Wächter, 2009).

² A página oficial com a documentação do *IPOPT* está disponível no endereço eletrônico: <https://github.com/coin-or/Ipopit>

4. ESTADO DA ARTE: O USO DE METAMODELOS PARA IMPLEMENTAÇÃO DA METODOLOGIA DE *SELF-OPTIMIZING CONTROL*

Nos últimos anos os estudos envolvendo a metodologia *self-optimizing control* têm sido direcionados na obtenção de estratégias de implementação precisas e que requerem um menor esforço computacional e de tempo enquanto mantêm a precisão dos resultados esperados. Dentro desse escopo, os metamodelos surgem como procedimentos matemáticos robustos que podem substituir o modelo do processo, que normalmente era representado por simuladores que possuem uma série de limitações, por exemplo, a necessidade de execução de muitas simulações que cresce exponencialmente com o tamanho do problema de *self-optimizing control*, a dificuldade de obtenção de gradientes e Hessianas de forma confiável, e o tempo de execução das simulações. Assim, o procedimento sistemático proposto nesse trabalho contribuirá no avanço e na discussão desse contexto do uso de metamodelos para implementação e estudos de *self-optimizing control*.

Nos últimos anos, Alves et al., (2018) desenvolveram uma sequência de procedimentos para resolução do problema de *self-optimizing control* usando metamodelos obtidos através de interpoladores *Kriging* para representar o modelo desenvolvido no simulador comercial *Aspen Plus*. Com os metamodelos, foi possível realizar a otimização do processo e a estimativa dos gradientes e Hessianas usando as funções matemáticas do interpolador *Kriging* considerando polinômio linear como regressor. Posteriormente, as variáveis controladas foram selecionadas para obter a menor perda média considerando erros de implementação e variações dos distúrbios. Todos os procedimentos e cálculos matemáticos foram desenvolvidos em *Matlab*. A metodologia foi aplicada para diferentes estudos de caso e os resultados validados com artigos da literatura. A metodologia proposta apresentou um avanço significativo na aplicação e desenvolvimento da metodologia de *self-optimizing control* por seus resultados precisos com um tempo de execução muito inferior quando comparado com os métodos tradicionais.

Como continuidade desse trabalho, Lima et al., (2020) implementaram a metodologia proposta por Alves et al., (2018) em linguagem *Python*. Para

implementação, algumas bibliotecas de cálculo foram desenvolvidas e estão atualmente disponibilizadas gratuitamente no *Github*. Para o planejamento e obtenção de metamodelos através de experimentos computacionais a biblioteca *pydace* foi implementada em linguagem *Python* (Lima et al., 2020) a partir da tradução parcial do modelo original do *DACE Toolbox* implementado em *Matlab* (Lophaven et al., 2002). A documentação e os detalhes de instalação e uso estão disponíveis gratuitamente na internet³. A implementação do procedimento de *self-optimizing control* é realizada utilizando os métodos discutidos na seção 3.5, para o Método Exato Local e o Método do Espaço Nulo Extendido. Os métodos foram implementados em linguagem *Python* (Lima et al., 2020) em uma biblioteca denominada *pysoc* disponível no *Github*⁴.

Além disso, uma ferramenta automatizada para o *Aspen Plus* também foi desenvolvida e disponibilizada gratuitamente para uma fácil utilização e resolução de problemas usando a metodologia *self-optimizing control*. A ferramenta foi nomeada *Metacontrol* e está disponível para instalação no *Github*⁵. Essa ferramenta é a primeira desenvolvida em linguagem aberta para implementação da metodologia *self-optimizing control* com modelos robustos e que não requer a instalação de *softwares* auxiliares (Lima et al., 2020).

Recentemente, Khezri et al., (2021) utilizaram metamodelos para implementar a metodologia de *self-optimizing control* para um processo de conversão de gás de síntese em combustíveis líquidos através da reação de Fischer-Tropsch. No trabalho quatro métodos para obtenção dos metamodelos foram utilizados: rede neural multicamada *perceptron* (MLP-ANN) usando regularização bayesiana (BR); redes neurais de função de base radial (RBF-ANN) usando o algoritmo genético (GA); máquina de suporte de vetor (SVM) usando otimização *particle swarm* (PSO); sistema de inferência neuro-fuzzy adaptativo (ANFIS) usando otimização *particle swarm* (PSO). Os metamodelos foram comparados em termos de prever o valor de saída e o ponto ótimo. O modelo metamodelo obtido pelo método MLP-ANN mostrou o melhor desempenho em quase todos os casos e foi considerado como representativo do simulador para o procedimento de *self-optimizing control*, obtendo um conjunto de variáveis controladas mais confiável que o procedimento clássico. Observa-se que nos últimos anos alguns trabalhos foram

³ <https://github.com/feslima/pydace>

⁴ <https://github.com/feslima/pysoc>

⁵ <https://github.com/feslima/metacontrol>

desenvolvidos avaliando a utilização de metamodelos para implementação da metodologia SOC com resultados promissores. Nesse sentido, esse trabalho tem o objetivo de propor uma metodologia para dar continuidade na discussão e viabilização do uso de metamodelos para resolução dos problemas de *self-optimizing control*.

5. IMPLEMENTAÇÃO DA METODOLOGIA

Nesse capítulo, a metodologia de *self-optimizing control* é implementada em linguagem *Python* através do ambiente de desenvolvimento *Jupyter Notebook*. As etapas são apresentadas conforme apresentado por Skogestad (2000a, 2000b) e detalhadas de maneira genérica para aplicação em diferentes modelos. As etapas para a implementação da metodologia SOC, estão descritas a seguir:

Etapa 1 – Análise dos graus de liberdade: *determine o número de graus de liberdade disponíveis para otimização, e identifique o conjunto de variáveis base para os graus de liberdade.*

Etapa 2 – Função objetivo e restrições: *defina o problema de operação ótima formulando uma função escalar custo para ser minimizada na operação ótima, e especifique as restrições que precisam ser satisfeitas durante a operação.*

Etapa 3 – Identifique os distúrbios mais importantes (aleatórios): *estes podem causar erros no modelo considerado para a otimização, distúrbios que ocorrem durante a operação e os erros de implementação para as variáveis controladas.*

Etapa 4 – Otimização: *resolver o problema de otimização definido na etapa 3.*

Etapa 5 – Identifique as candidatas a variáveis controladas: *entendimento sobre o processo e experiência, ou seja, conhecimento das variáveis que são dominantes, pode ser importante nesse estágio, porque o possível número de variáveis, e especialmente a combinação de variáveis, pode ser extremamente grande.*

Etapa 6 – Avaliação da perda: *calcule o valor médio da perda para as alternativas dos conjuntos de variáveis controladas.*

Etapa 7 – Outras análises e seleção: *selecione para outras análises o conjunto de variáveis controladas com perda aceitável (e as quais levam o sistema para o self-optimizing control).*

As Etapas 1 a 3 fazem parte do problema conceitual que deve ser estruturado de acordo com os objetivos desejados para o processo em estudo. Em nível

empresarial, os objetivos podem ser estabelecidos com relação às metas estabelecidas pela gerência, que normalmente são objetivos econômicos.

Especificamente para o objeto de estudo desse trabalho, iremos considerar que o problema é conhecido e as Etapas 1 a 3 já estão definidas. Para a Etapa 4 é necessário que o modelo do processo seja otimizado, modelo este que pode ser construído em linguagem de programação ou através de simuladores de processo. No entanto, previamente à execução da Etapa 4, é necessário desenvolver uma sequência de procedimentos para obtenção dos metamodelos que irão substituir e representar os modelos originais. Na seção 5.1 apresentaremos as etapas para o planejamento, metamodelagem e otimização do processo.

5.1. PLANEJAMENTO, METAMODELAGEM E OTIMIZAÇÃO

Inicialmente, o processo é modelado usando linguagens de programação ou simuladores de processo. Nesse trabalho, dois estudos de caso serão analisados, o primeiro estudo o modelo do processo será implementado em *Python* e no segundo caso no simulador de processos comercial *Aspen Hysys*. Em ambos os casos, todos os procedimentos matemáticos e de cálculos serão desenvolvidos em linguagem *Python* através do ambiente de desenvolvimento *Jupyter Notebook*. A escolha da linguagem de programação e do ambiente de desenvolvimento é devido a linguagem de programação ser gratuita e *open-source*, e a facilidade de modificação e interação mais com o usuário, respectivamente. Especificamente para o modelo desenvolvido em *Aspen Hysys*, a comunicação entre o *software* é realizada via linha de código usando a biblioteca *win32com* que é baseada em comunicação *COM* (*Component Object Model*). A Figura 2 apresenta o código de comunicação entre o *Python* e o *Aspen Hysys*.

```
In [ ]: # Comunicação com o Aspen Hysys
import win32com.client as win32
Hysys = win32.Dispatch('Hysys.Application')
sim_path = r'C:\Users\Folder\nome_do_arquivo.hsc'
MySimCase = Hysys.SimulationCases.Open(os.path.abspath(sim_path))
```

Figura 2. Código de comunicação entre o *Python* e o *Aspen Hysys*.

Após a construção dos modelos, faz-se necessário a estruturação de um planejamento de experimentos, onde as variáveis manipuladas e distúrbios são modificadas para obtenção dos dados de saída do modelo. É importante destacar que na proposta desse trabalho os experimentos realizam a amostragem das variáveis manipuladas (ou graus de liberdade) e dos distúrbios. Essa abordagem difere do trabalho desenvolvido por Alves et al., (2018) que realizava diferentes amostragens, inicialmente para os graus de liberdade e após o controle das variáveis ativas um novo planejamento de experimentos é realizado para o espaço reduzido (após o consumo de graus de liberdade para o controle das variáveis ativas). Neste trabalho a proposta é obter metamodelos que levam em consideração todas as variáveis que podem mudar ao longo do processo (manipuladas ou distúrbios) em relação às variáveis candidatas a controladas. Dessa forma, a comunicação com o modelo original só é realizada para a realização de um único e robusto planejamento de experimentos, diminuindo o tempo de execução de toda a metodologia de SOC.

Esses dados de entrada e saída são utilizados para o treinamento do metamodelo. Existem diversas técnicas de amostragem que podem ser utilizadas no planejamento de experimentos, no entanto, no desenvolvimento desse trabalho utilizaremos a técnica de amostragem *Latin Hypercube Sampling*, porém, o usuário tem a flexibilidade de utilizar outras técnicas de amostragem. O simulador *Aspen Hysys* apresenta uma funcionalidade interna denominada *Case Studies* que realiza a amostragem do modelo de acordo com perturbações pré-estabelecidas pelo usuário para as variáveis de entrada selecionadas. Para mais detalhes sobre essa técnica de amostragem é recomendado consultar os manuais técnicos do simulador.

Após o planejamento dos experimentos, o modelo do processo recebe uma matriz de dados das variáveis de entrada e executa cada ponto do planejamento, onde as variáveis manipuladas e distúrbios são modificadas e os valores das variáveis medidas são coletados. Para a modificação das variáveis de entrada no *Aspen Hysys* é necessário que as *tags* de cada variável sejam mapeadas a partir do modelo construído pelo usuário. Para isso, desenvolvemos um código de comunicação que recebe o arquivo de simulação do usuário e carrega todas as variáveis de entrada e saída de acordo com um dicionário criado para os principais blocos de modelos no simulador. Esse dicionário pode ser modificado a critério do

usuário para incluir novas variáveis consideradas críticas para determinado problema.

A etapa de planejamento de experimentos é considerada crítica na nossa proposta. O objetivo é obter um metamodelo robusto e confiável para o processo que irá resultar em ganho de tempo de computação nas etapas posteriores da metodologia *self-optimizing control*. Para isso, o número de pontos do planejamento é uma variável crítica para a construção de um metamodelo representativo. Para os simuladores de processo, a tolerância de resolução é uma variável importante que deve ser determinada após uma análise crítica do usuário para cada tipo de variável, tendo em vista que essa tolerância pode afetar diretamente a robustez do metamodelo em estimar os dados do processo influenciando diretamente nos cálculos das derivadas.

A partir dos dados de entrada e saída interpoladores *Kriging* são usados para obtenção dos metamodelos para as variáveis de saída e para a função objetivo. Nesse trabalho, utilizaremos um polinômio de regressão $f(x)$ de segunda ordem, conforme apresentado na Equação (68). Para implementação do *Kriging* utilizamos a biblioteca *pydace* que está disponível gratuitamente na internet.

Com os metamodelos das variáveis medidas e a função objetivo, a otimização do problema é realizada para determinação do ponto ótimo nominal da planta e verificação da existência de restrições ativas. O método de otimização utilizado é o *Ipopt (Interior Point Optimizer)* que é um método *open source* para otimização não-linear de larga escala. Para utilização em *Python*, a biblioteca *cyipopt* deve ser utilizada, que consiste em um *Python wrapper* escrito em *Cython*, que é um compilador que suporta chamadas de funções escritas em C. Detalhes específicos podem ser encontrados na documentação disponível de forma gratuita na internet⁶. Nesta etapa, o ponto ótimo obtido pelo metamodelo deve ser comparado com os resultados do modelo original para validação. Se os resultados diferem significativamente do modelo original, um novo planejamento de experimentos com mais pontos é necessário para o treinamento dos metamodelos.

Com o ponto ótimo calculado e validado com o modelo original, os gradientes e Hessianas podem ser estimados utilizando os metamodelos. Diferentemente do trabalho desenvolvido por Alves et al., (2018) que utiliza polinômio de regressão

⁶ A documentação para o *cyipopt* está disponível no endereço eletrônico:
<https://cyipopt.readthedocs.io/en/stable/index.html>

$f(x)$ de primeira ordem, nesse trabalho utilizamos os metamodelos obtidos a partir de polinômio de regressão de segunda ordem para estimativa dos gradientes e Hessianas. A utilização de polinômios de regressão de segunda ordem para os cálculos de gradientes e Hessianas influencia diretamente na redução do número de pontos necessários para obtenção de metamodelos representativos do processo. À medida que o número de pontos aumenta, os gradientes e Hessianas calculados tendem a estabilizar, nesse caso, essa observação significa que o metamodelo é matematicamente representativo do modelo original. Como o objetivo é a obtenção de gradientes e Hessianas precisas, os valores devem ser comparados para cada treinamento considerando os conjuntos de pontos.

Neste trabalho, a validação do metamodelo foi realizada através de critérios qualitativos de acordo com o conhecimento prévio das variáveis do modelo, no entanto, é recomendado que diferentes técnicas de validação e de inclusão de pontos sejam utilizadas para melhoria dos metamodelos. A Figura 3 apresenta o fluxograma das etapas descritas anteriormente para obtenção dos metamodelos validados e as matrizes de gradientes e Hessianas necessárias para a etapa de seleção das variáveis controladas. É importante destacar que não existe um número de pontos ideal para se obter bons metamodelos, logo, é necessário um procedimento iterativo e a comparação com o modelo original. Da mesma forma, quando estamos estimando gradientes e Hessianas, que são procedimentos matemáticos de aproximação, é necessário que estes permaneçam sem variações entre os conjuntos de dados de treinamentos. Ao final, obtemos metamodelos robustos e representativos do modelo original e as estimativas precisas dos gradientes e Hessianas, que influenciam diretamente no cálculo das perdas para diferentes conjuntos de variáveis controladas.

Na seção 5.2 discutiremos a implementação das Etapas 5 e 6 para determinação das melhores variáveis controladas e as respectivas perdas em relação ao processo.

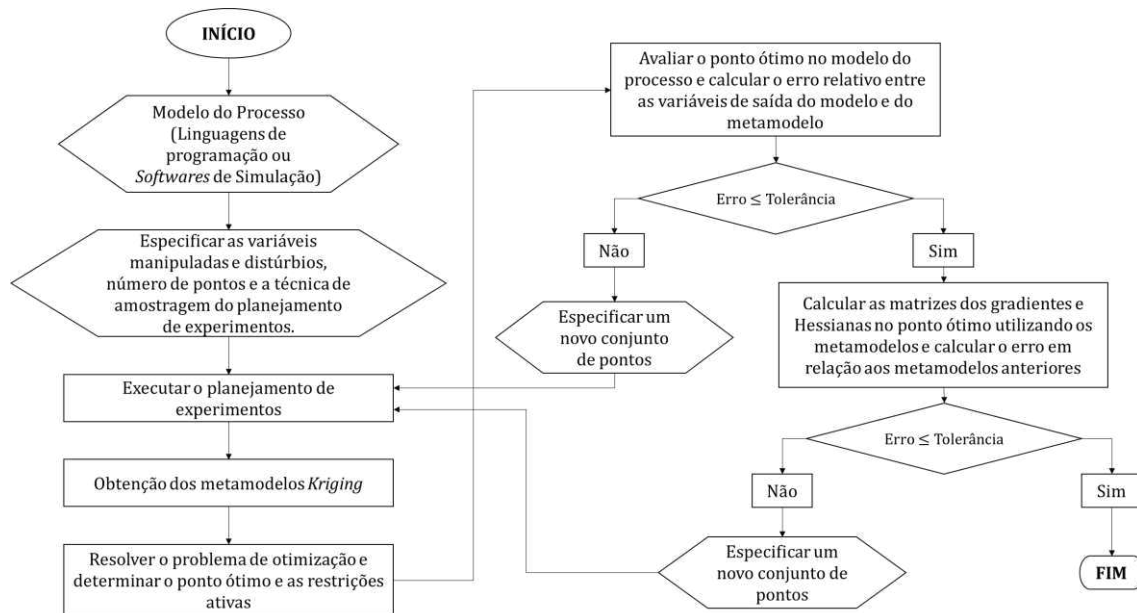


Figura 3. Fluxograma para o algoritmo de planejamento metamodelagem e otimização.

5.2. SELEÇÃO DAS VARIÁVEIS CONTROLADAS

A seleção das variáveis controladas que irão resultar na melhor perda são determinadas a partir da modelagem matemática e métodos descritos na Seção 3. A partir das estimativas dos gradientes e Hessianas, a metodologia do *self-optimizing control* pode ser aplicada. A metodologia SOC foi implementada em linguagem *Python* e está disponível no *Github*. A biblioteca *pysoc* disponibiliza o Método Exato Local e o Método do Espaço Nulo Estendido.

Como consideração no desenvolvimento da metodologia proposta por Skogestad (2000a), as variáveis ativas obtidas na otimização devem ser controladas. As variáveis ativas são obtidas durante a etapa de otimização. Os graus de liberdade restantes serão utilizados para determinar quais variáveis medidas devem ser controladas para obtenção da menor perda, quando mantidos os *setpoints* constantes.

Em alguns problemas, o número de variáveis medidas e candidatas a variáveis controladas disponíveis é maior que o número de graus de liberdade. Nesses casos, para determinar as melhores variáveis ou as suas combinações lineares, um grande número de combinações deve ser analisado para as possíveis

estruturas de controle. Para solucionar esse problema, o método do *branch and bound* proposto por Kariwala e Cao (2009) é usado para selecionar entre as melhores variáveis usando a perda média e a pior perda como métrica. A rotina também está disponível na biblioteca *pysoc* mencionada anteriormente.

Após a determinação das melhores variáveis controladas, é possível determinar os melhores pareamentos entre as variáveis manipuladas e controladas para os sistemas de controle utilizando a técnica *Relative Gain Array (RGA)*. A Figura 4 apresenta o algoritmo para as etapas de seleção das variáveis controladas.

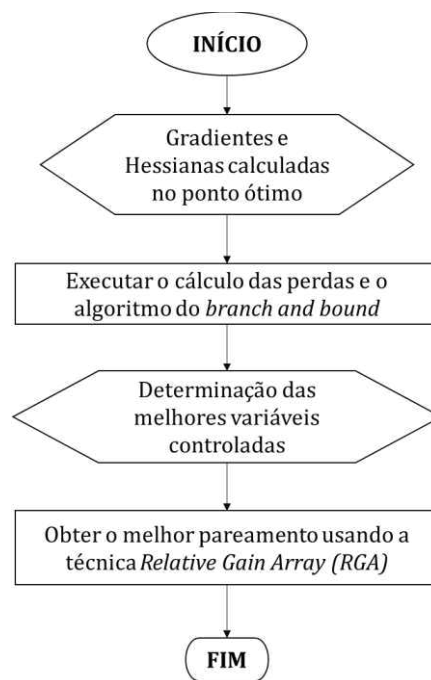


Figura 4. Fluxograma para o algoritmo de seleção das variáveis controladas.

Nas próximas seções, o procedimento proposto e a ferramenta implementada são aplicados para alguns estudos de caso e os resultados serão apresentados.

6. ESTUDOS DE CASO

Neste capítulo serão apresentados estudos de caso com a aplicação da metodologia proposta neste trabalho. Na seção 6.1 apresentaremos o problema de um reator CSTR com reação exotérmica modelado no *Python*. Na seção 6.2 apresentaremos um problema de uma coluna de destilação de alta pureza modelada no *Aspen Hysys*.

6.1. ESTUDO DE CASO 01: CSTR COM REAÇÃO REVERSÍVEL

Esse estudo de caso apresenta um reator continuamente agitado (CSTR) com uma reação reversível baseado no estudo de Economou et al., (1986). O processo consiste em um reator ideal continuamente agitado (Figura 5) onde acontece a seguinte reação exotérmica reversível:



com a taxa de reação sendo expressa por:

$$r = k_1 C_A - k_2 C_B \quad (75)$$

onde, $k_1 = k_{0,1} e^{\frac{-E_1}{RT}}$ e $k_2 = k_{0,2} e^{\frac{-E_2}{RT}}$. O modelo do processo para esse sistema é apresentado a seguir:

Balancos de Massa

$$\frac{dC_A}{dt} = \frac{1}{\tau} (C_{A,i} - C_A) - r \quad (76)$$

$$\frac{dC_B}{dt} = \frac{1}{\tau} (C_{B,i} - C_B) + r \quad (77)$$

Balanco de Energia

$$\frac{dT}{dt} = \frac{1}{\tau} (T_i - T) + \frac{-\Delta H_{rx}}{\rho C_p} r \quad (78)$$

Onde, $C_{j,i}$ é a concentração molar do componente j na corrente de entrada; C_j é a concentração molar no reator para o componente j ; T_i é a temperatura da corrente de entrada; T é a temperatura no reator; τ é o tempo de residência.

Os dados para o caso nominal e apresentado na Tabela 1.

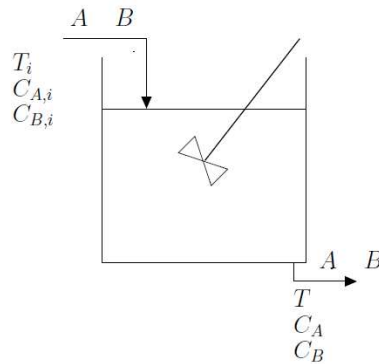


Figure 5. Ilustração do CSTR. (Fonte: Alstad, 2005).

Tabela 1. Dados nominais para o CSTR.

Parâmetros	Valor	Unidades
$k_{0,1}$	5000	s^{-1}
$k_{0,2}$	10^6	s^{-1}
C_p	1000	$cal\ kg^{-1}K^{-1}$
E_1	10^4	$cal\ mol^{-1}$
E_2	15000	$cal\ mol^{-1}$
R	1,987	$cal\ mol^{-1}K^{-1}$
T_i	400	K
$C_{A,i}$	1	$mol\ L^{-1}$
$C_{B,i}$	0	$mol\ L^{-1}$
$-\Delta H_{rx}$	5000	$cal\ mol^{-1}$
ρ	1	$kg\ L^{-1}$
τ	60	s

Esse problema de *self-optimizing* foi primeiramente proposto por Alstad (2005) considerando uma função econômica como objetivo de operação para maximizar o lucro, J , conforme apresentado na Equação (79).

$$J = [p_{C_B} C_B - (p_{T_i} T_i)^2] \quad (79)$$

Onde, $p_{C_B} = 2,009$ é o preço do produto C_B e $p_{T_i} = 1,657 \times 10^{-3}$ o custo de utilidade de aquecimento para a corrente de entrada do sistema.

A variável manipulada é $u = T_i$, temperatura da corrente de entrada, os estados do sistema são $\mathbf{x}^T = [C_A \ C_B \ T]$, e os distúrbios não medidos considerados são $\mathbf{d}^T = [C_{Ai} \ C_{Bi}]$. As variáveis medidas disponíveis são $\mathbf{y}^T = [C_A \ C_B \ T \ T_i]$.

A variação esperada para os distúrbios é $0,7 \leq C_{Ai} \leq 1,3$ e $C_{Bi} \leq 0,3$. O caso com $C_{Ai} = 1 \text{ mol/L}$ e $C_{Bi} = 0 \text{ mol/L}$ é considerado o ponto operacional nominal.

6.2. ESTUDO DE CASO 02: DESTILAÇÃO DE ALTA PUREZA PROPENO/PROPANO

Nesse estudo de caso uma coluna de destilação de alta pureza é utilizada para recuperar propeno como produto de topo com uma fração molar maior ou igual a 99,5%. Uma coluna de destilação similar foi analisada por Skogestad (2000a) e Alves, et al., (2018).

A corrente de alimentação é composta por uma mistura de propano e propeno com uma fração molar de propeno de 65%. A coluna tem 110 estágios incluindo condensador e refeedor. A corrente de alimentação entra na coluna como líquido saturado a uma pressão de 12 atm e uma vazão molar de 1 kmol. min^{-1} . Nominalmente, a coluna opera com uma vazão de destilado de $0,636 \text{ kmol. min}^{-1}$ e o fluxo de vapor de *boilup* de $13,90 \text{ kmol. min}^{-1}$. A alimentação entra no estágio 39 (contando a partir da base e considerando o refeedor como estágio 1), e uma queda de pressão de 1 atm é considerada ao longo da torre, com um estágio de top em 9,7 atm. O produto de topo deve possuir no mínimo 99,5% de propeno em fração molar. A única limitação física considerada no problema e imposta pelo sistema, devido a disponibilidade de energia, é a vazão máxima de vapor de *boilup* de $15 \text{ kmol. min}^{-1}$.

Entretanto, a torre tem dois graus de liberdade disponíveis para otimização. Os graus de liberdade (u) foram escolhidos como as especificações mais comuns para as operações de destilação, os quais são a vazão de destilado (D) e a vazão de vapor de *boilup* (V).

$$u = [D \ V]$$

Conforme explicitado anteriormente, a vazão de vapor de *boilup* possui como limitação física uma vazão máxima, e a fração de molar de propeno no topo (x_D), logo, temos as restrições:

$$x_D \geq 0,995$$

$$V \leq 15 \text{ kmol. min}^{-1}$$

Os distúrbios considerados nesse estudo de caso são os mesmos do estudo do Skogestad (2000a), a vazão molar de alimentação (F), a fração de vapor na corrente de alimentação (θ_F), e a composição de propeno na alimentação (z_F). As candidatas a variáveis controladas são as mesmo analisadas nos trabalhos de Skogestad (2000a) e Alves et al., (2018), a fração molar de propeno no destilado (x_D) a fração molar de propeno na base (x_B), a razão de destilado e carga (D/F), a razão de vazão de refluxo e carga (L/F), a razão de vapor de *boilup* e carga (V/F), e a razão de refluxo (L/D).

7. RESULTADOS E DISCUSSÕES

Nesta seção serão apresentados o desenvolvimento e resultados observados estudos de caso com a aplicação da metodologia proposta neste trabalho. O detalhamento e discussão dos resultados serão apresentados na seção 7.1 para o problema de um reator CSTR com reação exotérmica modelado no *Python* e na seção 7.2 apresentaremos um problema de uma coluna de destilação de alta pureza modelada no *Aspen Hysys*.

7.1 ESTUDO DE CASO 01: CSTR COM REAÇÃO REVERSÍVEL

O estudo de caso descrito na seção 6.1 foi analisado utilizando a metodologia proposta neste trabalho. Inicialmente, as equações do modelo foram implementadas no *Jupyter Notebook* e o problema de otimização foi resolvido usando a classe *optimize* e a função *fsolve* da biblioteca *scipy*.

A partir desse modelo, uma sequência de etapas foi conduzida com o objetivo de obter um metamodelo representativo para o estudo de caso em análise. Estas etapas seguem o procedimento descrito na seção 5.1 e estão descritas a seguir.

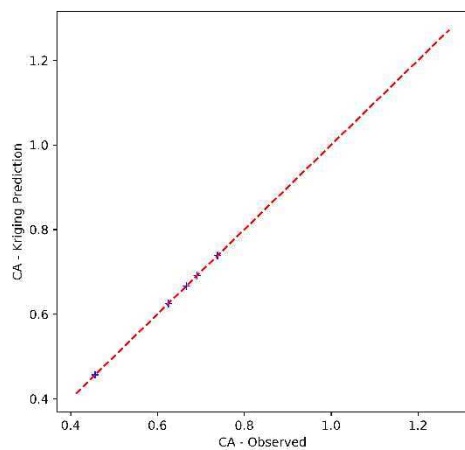
O planejamento de experimentos foi construído utilizando o método de amostragem *Latin Hypercube Sampling (LHS)* disponível na biblioteca *pyDOE2*. Conforme descrito na metodologia descrita na Figura 3, a amostragem foi realizada para as variáveis manipuladas e os distúrbios, representados por $input = [T_i \ C_{Ai} \ C_{Bi}]$. Para o *LHS*, é necessário fornecer os intervalos inferiores $lb = [350 \ 0,7 \ 0]$, e os intervalos superiores $ub = [500 \ 1,3 \ 0,3]$, respectivamente para cada variável. A escolha desses limites é realizada de acordo com o usuário baseada no conhecimento do processo. Para a obtenção de um metamodelo representativo do CSTR, o número de pontos foi avaliado até a obtenção de gradientes e Hessianas estáveis, utilizando o polinômio regressor de segunda ordem.

Para o problema, um total de 100 pontos foram utilizados para obter um metamodelo representativo. Para as variáveis medidas, consideramos $output = [C_A \ C_B \ T \ T_i]$. As variáveis *output*, para esse estudo de caso, são também consideradas como variáveis candidatas a controladas, y , que serão analisadas

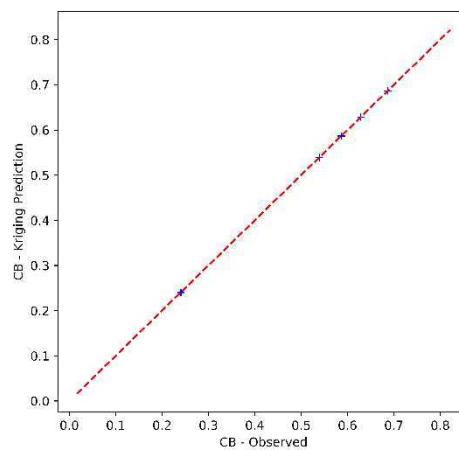
através da metodologia para determinação das melhores variáveis que devem ser controladas para obtenção do *self-optimizing control*. Os dados das variáveis *input* e *y* foram armazenados em uma matriz para o treinamento dos metamodelos.

Com os dados de *input/output*, interpoladores *Kriging* e polinômios regressores de segunda ordem foram utilizados considerando a biblioteca *pydace* modificada neste trabalho para o cálculo das Hessianas.

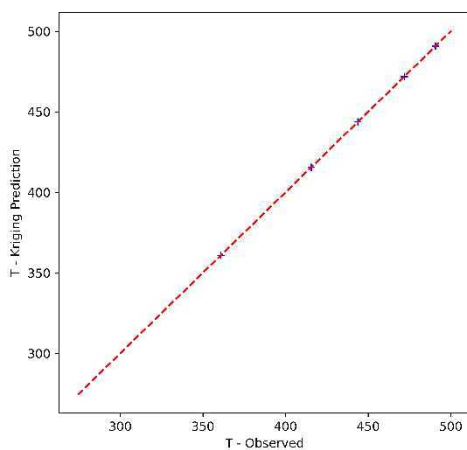
Para validação e avaliação dos metamodelos, 10 pontos foram analisados e estão apresentados na Figura 6 com a comparação entre os pontos calculados pelo modelo original e pelo metamodelo. A presença dos pontos sobre a reta indica que o metamodelo é representativo do modelo original. Um metamodelo também foi obtido para a função objetivo J para ser posteriormente utilizado para otimização. A Figura 7 apresenta a comparação da função objetivo entre o modelo original e o metamodelo.



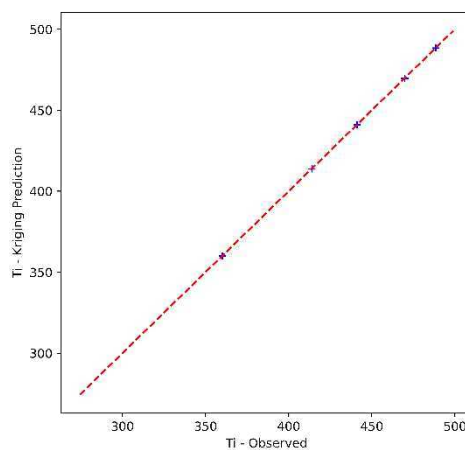
(a)



(b)



(c)



(d)

Figure 6. Validação dos metamodelos para as variáveis de saída do CSTR. (a) Concentração do reagente A; (b) Concentração do reagente B; (c) Temperatura do reator; (d) Temperatura da corrente de entrada do reator.

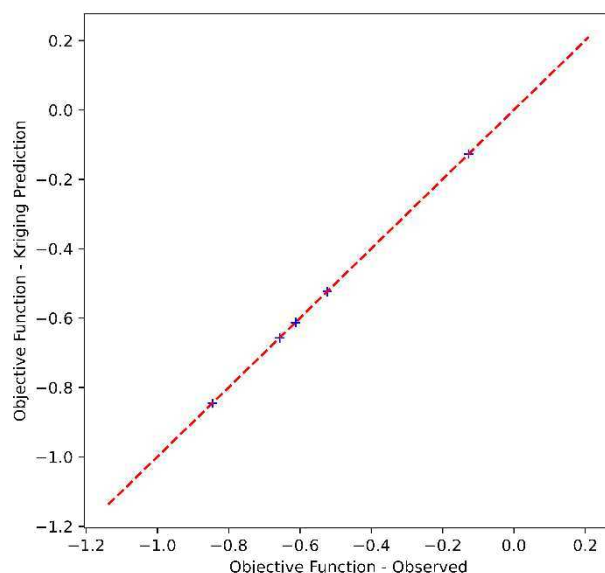


Figure 7. Validação dos metamodelos para a função objetivo.

Com os metamodelos, a otimização do processo é realizada para o caso nominal ($C_{Ai} = 1 \text{ mol. L}^{-1}$ e $C_{Bi} = 0 \text{ mol. L}^{-1}$), para a variação concentração inicial do componente A ($C_{Ai} = 0,7 \text{ mol. L}^{-1}$ e $C_{Bi} = 0 \text{ mol. L}^{-1}$) e ($C_{Ai} = 1,3 \text{ mol. L}^{-1}$ e $C_{Bi} = 0 \text{ mol. L}^{-1}$), e para a variação concentração inicial do componente B ($C_{Ai} = 1 \text{ mol. L}^{-1}$ e $C_{Bi} = 0,3 \text{ mol. L}^{-1}$). Em todos os casos, o objetivo é observar a variação do ponto ótimo e da função objetivo quando o modelo possui distúrbios. O problema de otimização é resolvido usando a biblioteca *cyipopt* e os resultados estão apresentados na Tabela 2 e comparados com os resultados obtidos por Alstad (2005). A Figura 8 apresenta o ponto ótimo e o seu desvio em relação ao caso nominal.

Tabela 2. Valores ótimos para diferentes distúrbios aplicados para os metamodelos do CSTR.

Variável	Alstad (2005)				Esse trabalho			
	Nominal	$C_{Ai} = 1,3$ mol. L^{-1}	$C_{Ai} = 0,7$ mol. L^{-1}	$C_{Bi} = 0,3$ mol. L^{-1}	nominal	$C_{Ai} = 1,3$ mol. L^{-1}	$C_{Ai} = 0,7$ mol. L^{-1}	$C_{Bi} = 0,3$ mol. L^{-1}
$C_A(\text{mol/L})$	0,498	0,644	0,352	0,585	0,498	0,644	0,352	0,586
$C_B(\text{mol/L})$	0,502	0,656	0,348	0,715	0,502	0,656	0,348	0,715
$T(K)$	426,761	429,170	422,601	415,883	426,803	429,208	422,639	415,924
$T_i(K)$	424,249	425,889	420,863	413,810	424,291	425,927	420,901	413,852
$-J(\$)$	0,515	0,821	0,212	0,966	0,515	0,820	0,212	0,965

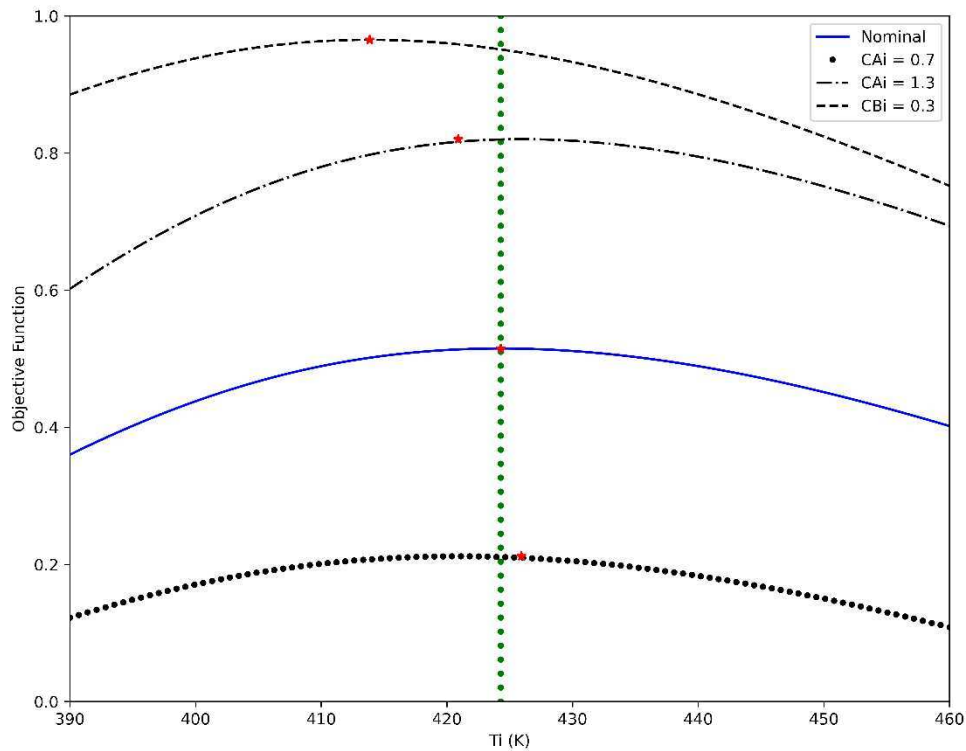


Figure 8. Função objetivo *versus* variável manipulada (T_i) para o caso nominal e diferentes distúrbios, os quais: $C_{Ai} = 0,7 \text{ mol.L}^{-1}$, $C_{Ai} = 1.3 \text{ mol.L}^{-1}$, $C_{Bi} = 0,3 \text{ mol.L}^{-1}$. A linha vertical em verde representa o ponto ótimo para o caso nominal.

Considerando o ponto ótimo obtido na otimização dos metamodelos, os gradientes e Hessianas devem ser calculados. A função *pydace* calcula os gradientes nativamente, no entanto, para o cálculo das Hessianas a expressão desenvolvida no estudo de Alves et al., (2018) foi utilizada para a derivação do regressor polinomial de segunda ordem. A implementação do cálculo da Hessiana faz parte desse trabalho e apresenta melhores resultados para o cálculo da Hessiana. Os resultados para os gradientes e Hessianas são comparados com os resultados de Kariwala (2007) obtidos analiticamente e são apresentados nas Equações (80) a (83). Os resultados apresentam uma estimativa consistente para a primeira e segunda derivadas usando metamodelos.

$$G^y_{Kariwala} = \begin{bmatrix} -0,001169 \\ 0,001169 \\ 1,005843 \\ 1 \end{bmatrix}, G^y_{calculado} = \begin{bmatrix} -0,00115967 \\ 0,00115967 \\ 1,00579837 \\ 1 \end{bmatrix} \quad (80)$$

$$G^y_{d_{Kariwala}} = \begin{bmatrix} 0,494719 & 0,277996 \\ 0,505281 & 0,722004 \\ 2,526403 & -1,389978 \\ 0 & 0 \end{bmatrix}'$$

$$G^y_{d_{calculado}} = \begin{bmatrix} 0,494723139 & 0,278014993 \\ 0,505276861 & 0,721985007 \\ 2,52638430 & -1,39007497 \\ 1,92150078 \times 10^{-13} & 2,83118502 \times 10^{-13} \end{bmatrix} \quad (81)$$

$$J_{uu_{Kariwala}} = [0,000234], \quad J_{uu_{calculado}} = [0,00023335] \quad (82)$$

$$J_{ud_{Kariwala}} = [-0,00177 \quad 0,008734],$$

$$J_{ud_{calculado}} = [-0,00177172 \quad 0,00873048] \quad (83)$$

O problema de *self-optimizing control* foi resolvido para determinar quais variáveis *output* = $[C_A, C_B, T, T_i]$ devem ser controladas para obter a menor perda operando com *setpoints* constantes sem a necessidade de otimização quando distúrbios acontecem no reator CSTR (Skogestad, 2000a). Primeiramente, seguindo a análise realizada por Kariwala (2007), os erros de implementação considerados para as medidas individuais foram considerados $0,01 \text{ mol.L}^{-1}$ para as concentrações C_A e C_B , e $0,5 \text{ K}$ para as medidas de temperatura T e T_i . Em relação aos distúrbios esperados, as magnitudes são $0,3 \text{ mol.L}^{-1}$ para ambas as concentrações C_{Ai} e C_{Bi} . A solução do problema considera o Método Exato Local com solução explícita proposta por Alstad et al., (2009) e Kariwala et al., (2008) que é aplicado para qualquer número de distúrbios e a solução minimiza não apenas a pior perda, mas também a perda média. Como é possível ter vários conjuntos ou combinações de variáveis controladas, o método do *branch and bound* proposto por Kariwala e Cao (2009) é usado para rapidamente selecionar entre os melhores conjuntos usando a perda média e a pior perda como métrica. A rotina *Python* usada para resolver o problema de *self-optimizing control* está disponível na biblioteca *pysoc* e aplicada nos trabalhos de Alves et al., (2018) e Lima et al., (2020). Os

resultados para o reator CSTR são apresentados na Tabela 3 e comparados com os resultados obtidos por Kariwala, (2007).

Tabela 3. Comparação das perdas para os casos mais promissores para controle do reator CSTR.

Ordem	Variáveis	Kariwala (2007)	Esse trabalho	
		Pior perda	Pior Perda	Perda Média
1	T_i	0,01532	0,01533	0,00170
2	T	0,016893	0,01691	0,00188
3	C_A	2,623395	2,658906	0,29543
4	C_B	5,58784	5,66787	0,62976

Como podemos observar a partir dos resultados apresentados na Tabela 3, a variável indicada pela metodologia SOC para ser controlada é a temperatura do reator, T , manipulando a temperatura da corrente de entrada, T_i . Observa-se também que a escolha das concentrações de saída leva o sistema a perdas significativas.

7.2. ESTUDO DE CASO 02: DESTILAÇÃO DE ALTA PUREZA PROPENO/PROPANO

O estudo de caso descrito na seção 6.2 foi analisado utilizando a metodologia proposta neste trabalho. Inicialmente, realizamos a comunicação da simulação desenvolvida em *Aspen Hysys*. As variáveis selecionadas foram automatizadas no *Jupyter Notebook* para executar o planejamento de experimentos. A Tabela 4 apresenta a lista de variáveis utilizadas para o planejamento de experimentos.

Tabela 4. Variáveis manipuladas, distúrbios e medidas e suas respectivas *tags* de automação no *Aspen Hysys*.

Variáveis	Classificação das variáveis	Tag no <i>Aspen Hysys</i>
D	Manipulada	"MyFlowsheet.Flowsheets.Item(0).ActiveSpecifications.Item(0).Goal"
V	Manipulada	"MyFlowsheet.Flowsheets.Item(0).ActiveSpecifications.Item(1).Goal"
θ_F	Distúrbio	"MyFlowsheet.MaterialStreams.Item(0).VapourFraction"
F	Distúrbio	"MyFlowsheet.MaterialStreams.Item(0).MolarFlow"
z_F	Distúrbio	"MyFlowsheet.MaterialStreams.Item(0).ComponentMolarFraction.Values[0]"
B	Medida	"MyFlowsheet.Flowsheets.Item(0).MaterialStreams.Item(5).MolarFlow.Value"
L	Medida	"MyFlowsheet.Flowsheets.Item(0).MaterialStreams.Item(0).MolarFlow.Value"
x_D	Medida	"MyFlowsheet.MaterialStreams.Item(1).ComponentMolarFraction.Values[0]"
x_B	Medida	"MyFlowsheet.MaterialStreams.Item(2).ComponentMolarFraction.Values[0]"

Após a automação e comunicação com o *Aspen Hysys*, uma série de etapas foram desenvolvidas para execução do procedimento apresentado neste trabalho e detalhado anteriormente na seção 5.1. As etapas estão detalhadas a seguir.

Um design de experimentos foi construído utilizando o método de amostragem *Latin Hypercube Sampling (LHS)* disponível na biblioteca *pyDOE2*. As variáveis manipuladas e distúrbios, representadas por $input = [D \ V \ \theta_F \ F \ z_F]$ foram amostradas para os intervalos inferiores $lb = [0,60 \ 14 \ 0 \ 1 \ 0,64]$, e os intervalos superiores $ub = [0,70 \ 15 \ 0,05 \ 1,05 \ 0,66]$, respectivamente.

É importante ressaltar um ponto crítico na obtenção de um bom planejamento de experimentos, para o caso em estudo, a simulação precisa ser ajustada para que as tolerâncias das variáveis manipuladas sejam significativas no planejamento de experimentos. O uso de tolerâncias que não sejam significativas em relação à ordem de magnitude das variáveis influencia na resolução da simulação. Nesse contexto, os pontos do planejamento poderão ter ordem de magnitudes menores que a tolerância, logo, os resultados seriam numericamente incoerentes. Dessa forma, magnitude de variação das variáveis dentre os pontos do planejamento de experimentos influenciam diretamente na obtenção do metamodelo.

Nesse estudo de caso em específico, observamos que para uma faixa de variação muito pequenas entre os limites inferiores e superiores das variáveis, os

pontos experimentais do *LHS* não apresentaram resultados matemáticos coerentes, e isso pode ser explicado pelo fato de o *Aspen Hysys* ser limitado pelas tolerâncias. Conforme já descrito ao longo do texto, o procedimento proposto neste trabalho consiste em obter um metamodelo representativo da coluna de destilação e o número de pontos foi avaliado até a obtenção de gradientes e Hessianas estáveis, utilizando o polinômio regressor de segunda ordem.

Para a coluna de destilação de alta pureza e as variáveis *input* amostradas no design de experimentos, um total de 1400 pontos foram necessários para obter um metamodelo representativo. Em relação às variáveis medidas que serão analisadas para determinação das variáveis controladas, consideramos $output = [D \ F \ V \ B \ L \ x_D \ x_B]$. Os dados das variáveis medidas foram armazenados em uma matriz para o treinamento dos metamodelos. No treinamento dos metamodelos é necessário especificar quais variáveis são candidatas a variáveis controladas para o problema de *self-optimizing control*, essas variáveis podem ser as variáveis medidas ou suas combinações matemáticas. Para a coluna de destilação, consideramos como variáveis candidatas a controladas $y = [x_B \ x_D \ L/F \ V/F \ L/D \ D/F]$. Os dados das variáveis *input* e y foram armazenados em uma matriz para o treinamento dos metamodelos.

Com os dados de *input/output*, interpoladores *Kriging* e polinômios regressores de segunda ordem foram utilizados considerando a biblioteca *pydace* modificada neste trabalho para o cálculo das Hessianas.

Nessa etapa é importante pontuar a importância do uso de regressores de segunda ordem para melhores estimativas dos metamodelos e dos gradientes e Hessianas. Análises foram realizadas para o estudo considerando apenas as duas variáveis manipuladas e os polinômios regressores de primeira ordem e foram necessários 1600 pontos para obtenção de metamodelos representativos, quando observamos os gradientes e as Hessianas. Para o problema com a variação de cinco variáveis (manipuladas e distúrbios), usando polinômios regressores de segunda ordem, 1400 pontos para suficientes para um modelo representativo.

Os metamodelos representativos foram otimizados para a função objetivo J conforme Equação (84) e as restrições descritas na seção 5.2.

$$J (\$/\text{min}) = -[20D + (10 - 20x_D)B - 10F - 0,3V] \quad (84)$$

$$x_D \geq 0,995$$

$$V \leq 15 \text{ kmol. min}^{-1}$$

Os resultados mostraram que os valores ótimos foi $D = 0,64332857 \text{ kmol. min}^{-1}$ e $V = 14,50108579 \text{ kmol. min}^{-1}$, com uma restrição ativa para a composição de propeno no destilado, $x_D = 0,995$. A Figura 9 apresenta o gráfico das curvas de nível para a função objetivo e a restrição $x_D = 0,995$. O problema de otimização é resolvido usando a biblioteca *cyipopt* e os resultados estão apresentados na Tabela 5, onde é possível observar que os metamodelos obtidos são representativos do modelo implementado em *Aspen Hysys*.

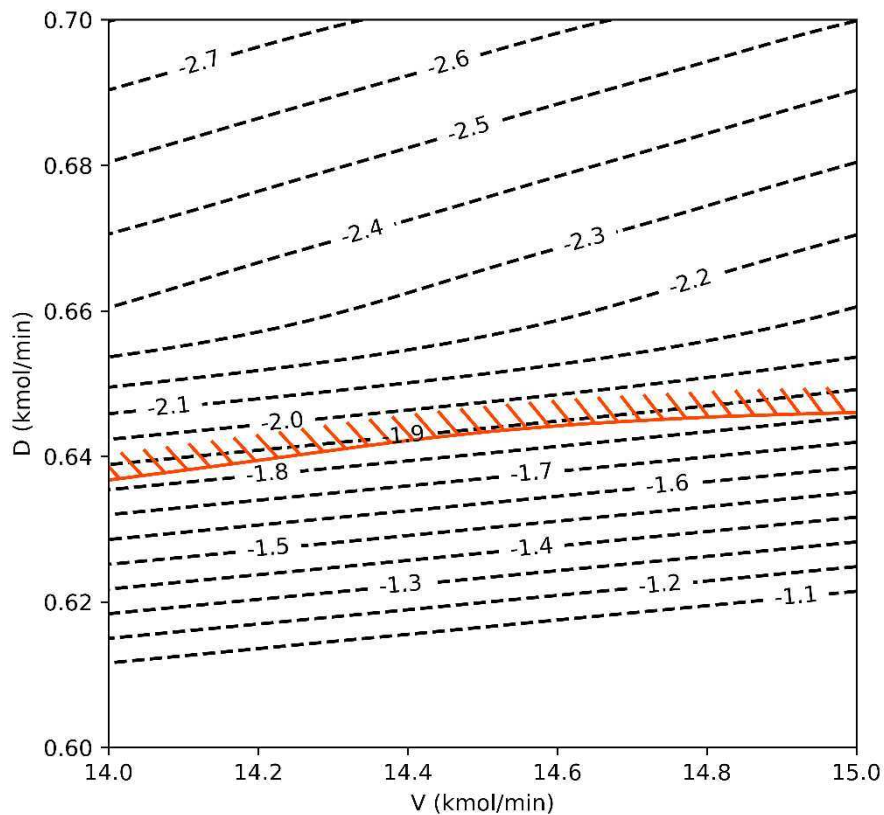


Figure 9. Curvas de nível para função objetivo em função das variáveis manipuladas. A restrição x_D é representada pela curva em vermelho.

Tabela 5. Resultados dos metamodelos e *Aspen Hysys* para o ponto ótimo.

Materiais	Metamodelo	Aspen Hysys
Função Objetivo (\$/min)	-1,885211	-1,885548
B (kmol/min)	0,356671	0,356666
x_B (-)	0,027719	0,027684
x_D (-)	0,994999	0,995016

Considerando o ponto ótimo obtido na etapa de otimização, os gradientes e Hessianas devem ser calculados. A função *pydace* calcula os gradientes nativamente, no entanto, para o cálculo das Hessianas o regressor polinomial de segunda ordem, os quais apresentam metamodelos obtidos com um menor número de pontos. As Equações (85) e (86) apresentam os resultados dos gradientes para as candidatas a variáveis controladas em relação às variáveis manipuladas (G_y) e aos distúrbios (G_y^d). As candidatas a variáveis controladas foram x_B , x_D , L/F , V/F , L/D e D/F . As Equações (87) e (88) apresentam as segundas derivadas da função objetivo em reação às variáveis manipuladas (J_{uu}) e aos distúrbios (J_{ud}). Esses resultados matemáticos são necessários para os cálculos da metodologia do *self-optimizing control*.

$$G_y = \begin{bmatrix} \frac{dx_B}{dD} & \frac{dx_B}{dV} \\ \frac{dx_D}{dD} & \frac{dx_D}{dV} \\ \frac{d(L/F)}{dD} & \frac{d(L/F)}{dV} \\ \frac{d(V/F)}{dD} & \frac{d(V/F)}{dV} \\ \frac{d(L/D)}{dD} & \frac{d(L/D)}{dV} \\ \frac{d(D/F)}{dD} & \frac{d(D/F)}{dV} \end{bmatrix} = \begin{bmatrix} -2,5494 & -0,00162894 \\ -0,090098 & 0,0009054264 \\ -0,86327 & 0,97709 \\ 4,3385 \times 10^{-8} & 1,0 \\ -0,34314 & 1,5181 \\ 1,0 & -1,8295 \times 10^{-11} \end{bmatrix} \quad (85)$$

$$G_y^d = \begin{bmatrix} \frac{dx_B}{d\theta_F} & \frac{dx_B}{dF} & \frac{dx_B}{dz_F} \\ \frac{dx_D}{d\theta_F} & \frac{dx_D}{dF} & \frac{dx_D}{dz_F} \\ \frac{d(L/F)}{d\theta_F} & \frac{d(L/F)}{dF} & \frac{d(L/F)}{dz_F} \\ \frac{d(V/F)}{d\theta_F} & \frac{d(V/F)}{dF} & \frac{d(V/F)}{dz_F} \\ \frac{d(L/D)}{d\theta_F} & \frac{d(L/D)}{dF} & \frac{d(L/D)}{dz_F} \\ \frac{d(D/F)}{d\theta_F} & \frac{d(D/F)}{dF} & \frac{d(D/F)}{dz_F} \end{bmatrix} = \begin{bmatrix} -0,00477483 & 1,5850 & 2,6290 \\ 0,00256483 & 0,086639 & 0,097148 \\ 0,93635 & -13,683 & -0,15798 \\ 4,9908 \times 10^{-9} & -14,501 & 6,2849 \times 10^{-8} \\ 1,4274 & -0,18089 & -0,30798 \\ 1,0622 \times 10^{-9} & -0,64332 & 4,2711 \times 10^{-10} \end{bmatrix} \quad (86)$$

$$J_{uu} = \begin{bmatrix} \frac{\partial J^2}{\partial D^2} & \frac{\partial J^2}{\partial V \partial D} \\ \frac{\partial J^2}{\partial D \partial V} & \frac{\partial J^2}{\partial V^2} \end{bmatrix} = \begin{bmatrix} 210,56 & -0,089695 \\ -0,089695 & 0,01024 \end{bmatrix} \quad (87)$$

$$J_{ud} = \begin{bmatrix} \frac{\partial J^2}{\partial D \partial \theta_F} & \frac{\partial J^2}{\partial D \partial F} & \frac{\partial J^2}{\partial D \partial z_F} \\ \frac{\partial J^2}{\partial V \partial \theta_F} & \frac{\partial J^2}{\partial V \partial F} & \frac{\partial J^2}{\partial V \partial z_F} \end{bmatrix} = \begin{bmatrix} -6,0638 & -45,203 & -115,33 \\ -0,013172 & 0,0012334 & 0,13302 \end{bmatrix} \quad (88)$$

Para o problema de *self-optimizing control* foram consideradas as magnitudes para a variação dos distúrbios de 0,15 para a fração de vapor saturado (θ_F), 0,20 para a vazão molar de alimentação (F) e 0,05 para a fração de propeno na alimentação (x_D). Para os erros de implementação das variáveis medidas e candidatas às variáveis controladas foram consideradas magnitudes de 0,0001 para as frações molares x_B e x_D , e 0,001 para as variáveis L/F , V/F , L/D e D/F . Esses valores são considerados para os cálculos da perda média e pior perda como as matrizes de magnitude dos distúrbios, W_d , e magnitude dos erros de implementação das variáveis medidas, W_n . Logo,

$$W_d = [0,15 \ 0,20 \ 0,05] \quad (89)$$

$$W_n = [0,0001 \ 0,0001 \ 0,001 \ 0,001 \ 0,001 \ 0,001] \quad (90)$$

Segundo a metodologia proposta por Skogestad (2000a), as restrições ativas devem ser controladas. Para o nosso problema, a fração molar de propeno no destilado é ativa no ponto ótimo ($x_D = 0,995$) e deve ser controlada. Dentre os graus de liberdade disponíveis, a variável selecionada foi D . A escolha da variável para controlar a variável ativa independe, porque a análise é realizada em estado estacionário. Esse resultado é facilmente verificável através da ferramenta. Logo, a metodologia do *self-optimizing control* pode ser aplicada para determinar dentre as variáveis candidatas a controladas qual apresenta menor perda quando

manipulamos a vazão de vapor de *boilup*, V . Os resultados estão apresentados na Tabela 6.

Tabela 6. Resultados das candidatas a variáveis controladas em ordem crescente das perdas.

Candidatas	Perda Média (\$/min)	Pior Média (\$/min)
L/D	0,000234	0,00281
L/F	0,003639	0,043662
V/F	0,003845	0,046145
x_D	0,165572	1,986863
x_B	18,991964	227,903572
D/F	$2,1 \times 10^{18}$	$2,5 \times 10^{17}$

De acordo com os resultados obtidos, podemos concluir que as variáveis que devem ser controladas, y_{CV} , para apresentar uma menor perda são $y_{CV} = [x_D \ L/D]$. O pareamento das variáveis controladas e manipuladas pode ser realizado através da técnica de RGA (*Relative Gain Array*) (Bristol, 1966), que considera as interações dinâmicas entre as variáveis. Aplicando a técnica de RGA obtemos os pares MV-CV para as variáveis resultado da metodologia do *self-optimizing control* $D - x_D$ e $V - L/D$, com pouca interação entre as variáveis controladas. A Matriz RGA é apresentada a seguir na Equação (91).

$$RGA = \begin{bmatrix} 1,29389738 & -0,29389738 \\ -0,29389738 & 1,29389738 \end{bmatrix} \quad (91)$$

8. CONTRIBUIÇÕES E CONCLUSÕES

Através desta pesquisa, foi possível explorar e analisar o desenvolvimento de uma ferramenta desenvolvida em linguagem *Python*, implementada no ambiente de desenvolvimento *Jupyter Notebook* para implementação da metodologia de *self-optimizing control* usando metamodelos *Kriging*. Os resultados obtidos revelaram que a ferramenta desenvolvida é aplicável para problemas já explorados na literatura com resultados coerentes. Essas descobertas contribuem significativamente para o conhecimento atual no campo de *self-optimizing control* e fornecem *insights* valiosos para aplicação da metodologia de maneira mais rápida e flexível para o usuário com robustez matemática.

Além disso, durante o processo de desenvolvimento, foi possível validar a aplicação da metodologia para modelos de processos construídos em *Python* (estudo de caso 01) e para modelos construídos no *software* de simulação *Aspen Hysys* (estudo de caso 02). Para o primeiro estudo de caso foi possível reproduzir os resultados obtidos na literatura com precisão, indicando que o procedimento proposto neste trabalho é aplicável para a metodologia *self-optimizing control*. Para o segundo estudo de caso, observações importantes foram observadas durante o desenvolvimento e aplicação do procedimento. Através dos estudos, observamos que é de crítica importância a escolha da tolerância usada no simulador de processo, especificamente para esse trabalho o *Aspen Hysys*. As tolerâncias influenciam diretamente no número de pontos necessários para obtenção de um metamodelo representativo do processo.

Ainda como contribuição significativa, nesse trabalho propusemos a modificação do cálculo das Hessianas usando metamodelos, reduzindo assim o número de pontos necessários para essas estimativas matemáticas quando comparado com os métodos já apresentados na literatura. Essas observações foram essenciais para a realização de uma análise abrangente e rigorosa, permitindo uma compreensão mais profunda do uso de metamodelos para aplicação da metodologia de *self-optimizing control*.

Como resultado dessa pesquisa, os resultados obtidos apoiam a ideia de que a ferramenta é funcional e pode ser explorada para problemas mais complexos e

outros problemas relacionados ao tema de *self-optimizing control*, por exemplo, para problemas de larga escala.

No entanto, é importante ressaltar que este estudo está em fase de desenvolvimento, sendo necessário o desenvolvimento de etapas complementares para melhorar a aplicabilidade e reprodutibilidade do procedimento.

9. SUGESTÕES DE TRABALHOS FUTUROS

A continuidade do desenvolvimento de ferramentas que possibilitem a aplicabilidade da metodologia de *self-optimizing control* é essencial para que a metodologia adentre os processos industriais reais, principalmente na fase de *design* e concepção. Tomando como base a proposta apresentada nesse trabalho, sugerem-se algumas modificações para melhoria da performance e funcionalidade da ferramenta:

- 1) Implementação de técnicas estatísticas de validação dos metamodelos em comparação ao modelo original. É importante que as técnicas de validação propostas levem em consideração a magnitude do valor das variáveis, que estão diretamente relacionadas com a tolerância do modelo, em especial os modelos construídos em simuladores de processo;
- 2) Implementação de técnicas de validação para os gradientes e Hessianas. Conforme discutido ao longo do trabalho, a precisão na obtenção dessas derivadas é crucial para a metodologia SOC. Em especial para as segundas derivadas, um maior número de pontos é necessário para obtenção dessas matrizes. Sugere-se que após a primeira validação do ponto ótimo em relação ao modelo original, novos pontos em torno do ótimo sejam inseridos no conjunto de dados de treinamento dos metamodelos;
- 3) Realizar a automatização das etapas de planejamento, metamodelagem e otimização para que o procedimento seja sistemático e com a mínima intervenção do usuário;
- 4) Aplicação da metodologia SOC através do procedimento proposto nesse trabalho para problemas onde todas as variáveis sejam ativas para determinação dos melhores pareamentos;
- 5) Implementação de outras técnicas de amostragem e de metamodelagem;
- 6) Automação da metodologia SOC para outros simuladores de processo ampliando as possibilidades de desenvolvimento de pesquisa em nível acadêmico e industrial;
- 7) Realizar testes da metodologia proposta para problemas mais complexos e com um maior número de variáveis.

10. REFERÊNCIAS BIBLIOGRÁFICAS

- ALSTAD, Vidar; SKOGESTAD, Sigurd. Combination of measurements as controlled variables for self-optimizing control. In: **Computer Aided Chemical Engineering**. Elsevier, p. 353-358, 2003.
- ALSTAD, Vidar. Studies on selection of controlled variables. **Norwegian University of Science and Technology**, 2005.
- ALSTAD, Vidar; SKOGESTAD, Sigurd. Null space method for selecting optimal measurement combinations as controlled variables. **Industrial & engineering chemistry research**, v. 46, n. 3, p. 846-853, 2007.
- ALSTAD, Vidar; SKOGESTAD, Sigurd; HORI, Eduardo S. Optimal measurement combinations as controlled variables. **Journal of Process Control**, v. 19, n. 1, p. 138-148, 2009.
- ALVES, Victor MC et al. Metamodel-based numerical techniques for self-optimizing control. **Industrial & Engineering Chemistry Research**, v. 57, n. 49, p. 16817-16840, 2018.
- BRISTOL, Edgar. On a new measure of interaction for multivariable process control. **IEEE transactions on automatic control**, v. 11, n. 1, p. 133-134, 1966.
- BUCKLEY, P. S. *Techniques of Process Control*; Wiley: New York, 1964.
- CAO, Yi; KARIWALA, Vinay. Bidirectional branch and bound for controlled variable selection: Part I. Principles and minimum singular value criterion. **Computers & Chemical Engineering**, v. 32, n. 10, p. 2306-2319, 2008.
- CHRISTINE, Ng; STEPHANOPOULOS, George. Plant-wide control structures and strategies. **IFAC Proceedings Volumes**, v. 31, n. 11, p. 1-16, 1998.
- DE ARAÚJO, Antonio CB; GOVATSMARK, Marius; SKOGESTAD, Sigurd. Application of plantwide control to the HDA process. I—steady-state optimization and self-optimizing control. **Control engineering practice**, v. 15, n. 10, p. 1222-1237, 2007.
- DE ARAUJO, Antonio CB et al. Sensitivity analysis of optimal operation of an activated sludge process model for economic controlled variable selection. **Industrial & Engineering Chemistry Research**, v. 52, n. 29, p. 9908-9921, 2013.
- ECONOMOU, Constantin G.; MORARI, Manfred; PALSSON, Bernhard O. Internal model control: Extension to nonlinear system. **Industrial & Engineering Chemistry Process Design and Development**, v. 25, n. 2, p. 403-411, 1986.
- FOSS, A. Critique of chemical process control theory. **IEEE Transactions on Automatic Control**, v. 18, n. 6, p. 642-652, 1973.
- GODOY, Rodrigo J. C.; GARCIA, Claudio. Plantwide control: a review of design techniques, benchmarks, and challenges. **Industrial & Engineering Chemistry Research**, v. 56, n. 28, p. 7877-7887, 2017.

- GRACIANO, José Eduardo A. et al. Integrating self-optimizing control and real-time optimization using zone control MPC. **Journal of Process Control**, v. 34, p. 35-48, 2015.
- HALVORSEN, Ivar J. et al. Optimal selection of controlled variables. **Industrial & Engineering Chemistry Research**, v. 42, n. 14, p. 3273-3284, 2003.
- JÄSCHKE, Johannes; SKOGESTAD, Sigurd. NCO tracking and self-optimizing control in the context of real-time optimization. **Journal of Process Control**, v. 21, n. 10, p. 1407-1416, 2011.
- JÄSCHKE, Johannes; CAO, Yi; KARIWALA, Vinay. Self-optimizing control—A survey. **Annual Reviews in Control**, v. 43, p. 199-223, 2017.
- KARIWALA, Vinay. Optimal measurement combination for local self-optimizing control. **Industrial & Engineering Chemistry Research**, v. 46, n. 11, p. 3629-3634, 2007.
- KARIWALA, Vinay; CAO, Yi; JANARDHANAN, S. Local self-optimizing control with average loss minimization. **Industrial & Engineering Chemistry Research**, v. 47, n. 4, p. 1150-1158, 2008.
- KARIWALA, Vinay; CAO, Yi. Bidirectional branch and bound for controlled variable selection. Part II: Exact local method for self-optimizing control. **Computers & chemical engineering**, v. 33, n. 8, p. 1402-1412, 2009.
- KARIWALA, Vinay; CAO, Yi. Bidirectional branch and bound for controlled variable selection part III: Local average loss minimization. **IEEE Transactions on Industrial Informatics**, v. 6, n. 1, p. 54-61, 2010.
- KHEZRI, Vahid et al. Application of surrogate models as an alternative to process simulation for implementation of the self-optimizing control procedure on large-scale process plants—A natural gas-to-liquids (GTL) case study. **Industrial & Engineering Chemistry Research**, v. 60, n. 13, p. 4919-4929, 2021.
- KRIGE, D. G. A statistical approach to some mine valuations and allied problems at the Witwatersrand. Master's thesis. University of Witwatersrand. South Africa. 1951.
- LARSSON, Truls; SKOGESTAD, Sigurd. Plantwide control - A review and a new design procedure. **Modeling, Identification and Control**, v. 21, n. 4, p. 209-240, 2000.
- LARSSON, Truls et al. Self-optimizing control of a large-scale plant: The Tennessee Eastman process. **Industrial & engineering chemistry research**, v. 40, n. 22, p. 4889-4901, 2001.
- LIMA, Felipe Souza; ALVES, Victor Manuel Cunha; ARAUJO, Antonio Carlos Brandao. Metacontrol: A Python based application for self-optimizing control using metamodels. **Computers & Chemical Engineering**, v. 140, p. 106979, 2020.
- LOPHAVEN, Søren Nymand et al. **DACE: a Matlab kriging toolbox**. Lyngby, Denmark: IMM, Informatics and Mathematical Modelling, The Technical University of Denmark, 2002.

MORARI, Manfred; ARKUN, Yaman; STEPHANOPOULOS, George. Studies in the synthesis of control structures for chemical processes: Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures. **AIChE Journal**, v. 26, n. 2, p. 220-232, 1980.

NOCEDAL, Jorge; WÄCHTER, Andreas; WALTZ, Richard A. Adaptive barrier update strategies for nonlinear interior methods. **SIAM Journal on Optimization**, v. 19, n. 4, p. 1674-1693, 2009.

SACKS, Jerome et al. Design and analysis of computer experiments. **Statistical science**, v. 4, n. 4, p. 409-423, 1989.

SILVA, S. K. et al. Development and application of an automatic tool for the selection of control variables based on the self-optimizing control methodology. **Brazilian Journal of Chemical Engineering**, v. 34, p. 851-871, 2017.

SKOGESTAD, Sigurd; HALVORSEN, Ivar J.; MORUD, John C. **Self-optimizing control: The basic idea and taylor series analysis**. American Institute of Chemical Engineers, 1998.

SKOGESTAD, Sigurd. Plantwide control: The search for the self-optimizing control structure. **Journal of process control**, v. 10, n. 5, p. 487-507, 2000a.

SKOGESTAD, Sigurd. Self-optimizing control: The missing link between steady-state optimization and control. **Computers & Chemical Engineering**, v. 24, n. 2-7, p. 569-575, 2000b.

SKOGESTAD, Sigurd. Control structure design for complete chemical plants. **Computers & Chemical Engineering**, v. 28, n. 1-2, p. 219-234, 2004.

SKOGESTAD, Sigurd. Economic plantwide control. **Plantwide Control: Recent developments and applications**, p. 229-251, 2012.

SKOGESTAD, S.; POSTLETHWAITE, I. **Multivariable feedback control**, 2nd edition. John Wiley & Sons, 2005.

STEPHANOPOULOS, George. Synthesis of control systems for chemical plants A challenge for creativity. **Computers & chemical engineering**, v. 7, n. 4, p. 331-365, 1983.

VAN DE WAL, Marc; DE JAGER, Bram. A review of methods for input/output selection. **Automatica**, v. 37, n. 4, p. 487-510, 2001.

WÄCHTER, Andreas. Short tutorial: Getting started with ipopt in 90 minutes. In: **Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik**, 2009.

WÄCHTER, Andreas. **An interior point algorithm for large-scale nonlinear optimization with applications in process engineering**. 2002. Tese de Doutorado. Carnegie Mellon University.

WÄCHTER, Andreas; BIEGLER, Lorenz T. Line search filter methods for nonlinear programming: Motivation and global convergence. **SIAM Journal on Optimization**, v. 16, n. 1, p. 1-31, 2005a.

WÄCHTER, Andreas; BIEGLER, Lorenz T. Line search filter methods for nonlinear programming: Local Convergence. **SIAM Journal on Optimization**, v. 16, n. 1, p. 32-48, 2005b.

WÄCHTER, Andreas; BIEGLER, Lorenz T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. **Mathematical programming**, v. 106, p. 25-57, 2006.

WILLIAMS, Christopher KI; RASMUSSEN, Carl Edward. **Gaussian processes for machine learning**. Cambridge, MA: MIT press, 2006.

APÊNDICE – A

Os arquivos modificados para o cálculo das Hessianas e os arquivos usados para os resultados obtidos neste trabalho podem ser solicitados através dos endereços de e-mail:

marcilio.maximo@eq.ufcg.edu.br

marcilio.ms1992@gmail.com

É importante destacar que essas modificações realizadas para as funções utilizadas para o cálculo das Hessianas utilizando metamodelos através de polinômios regressores de segunda ordem foram realizadas nos arquivos internos do computador onde a metodologia é executada, pois, a biblioteca *pydace* não é gerenciada pelos autores desse trabalho.