

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
Escola de Engenharia  
Programa de Pós-Graduação em Engenharia Elétrica

Victor São Paulo Ruela

**INVESTIGAÇÃO DE ABORDAGENS EVOLUTIVAS E OTIMIZAÇÃO  
BAYESIANA RESTRITA NO PROBLEMA DE OTIMIZAÇÃO EM  
TEMPO-REAL**

Belo Horizonte  
2022

Victor São Paulo Ruela

**INVESTIGAÇÃO DE ABORDAGENS EVOLUTIVAS E OTIMIZAÇÃO  
BAYESIANA RESTRITA NO PROBLEMA DE OTIMIZAÇÃO EM  
TEMPO-REAL**

**Versão Final**

Dissertação enviada ao Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, em cumprimento parcial dos requisitos para o título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Michel Bessani

Belo Horizonte  
2022

R921i

Ruela, Victor São Paulo.

Investigação de abordagens evolutivas e otimização Bayesiana restrita no problema de otimização em tempo-real [recurso eletrônico] / Victor São Paulo Ruela. - 2021.

1 recurso online (86 f. : il., color.) : pdf.

Orientador: Michel Bessani.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f.79-86.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia Elétrica - Teses. 2. Processos Gaussianos – Teses.  
3. Teoria bayesiana de decisão estatística – Teses. I. Bessani, Michel.  
II. Universidade Federal de Minas Gerais. Escola de Engenharia.  
III. Título.

CDU: 621.3(043)

**"Investigação de Abordagens Evolutivas e Otimização Bayesiana Restrita No Problema de Otimização Em Tempo-real"**

**Victor São Paulo Ruela**

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

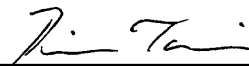
Aprovada em 16 de dezembro de 2021.

Por:



---

**Prof. Dr. Michel Bessani**  
DEE (UFMG) – Orientador



---

**Prof. Dr. Ricardo Hiroshi Caldeira Takahashi**  
DMAT (UFMG)



---

**Prof. Dr. Esly Ferreira da Costa Junior**  
DEQ (UFMG)

*Este trabalho é dedicado àqueles que acordam todos os dias  
antes do sol nascer para correr atrás dos seus sonhos.*

# Agradecimentos

Agradeço aos meus pais, Otaviano e Juliana, por terem me dado suporte incondicional durante toda minha vida acadêmica.

Ao meu orientador Professor Michel Bessani, por ter acreditado nas minhas ideias e sempre me incentivado à inovar e explorar todo o meu potencial.

A todos os amigos do ciclismo, serei eternamente grato pelos momentos compartilhados, os quais foram essenciais para garantir o equilíbrio necessário para esta jornada.

À Nivia pela paciência e carinho nos momentos mais difíceis da execução deste trabalho.

A todos os colegas de trabalho que contribuíram direta ou indiretamente para o meu desenvolvimento profissional e pessoal, indispensáveis para a conclusão deste ciclo.

*“It never gets easier, you just go faster“  
(LeMond, Greg)*

# Resumo

A otimização em tempo-real (RTO) é uma técnica capaz de iterativamente levar um processo industrial (planta) à sua operação econômica ótima, utilizando para isso um modelo matemático aproximado aliado à solução de um problema de otimização não linear. Para lidar com discrepâncias entre o modelo e a planta estão disponíveis na literatura diferentes abordagens, entre as quais a de *Modifier Adaptation* (MA) se destaca. Ela aplica correções de primeira ordem às funções de custo e restrições a fim de alcançar o ótimo da planta após a convergência. No entanto, o cálculo das correções depende de informações do gradiente da planta, que são difíceis de obter. Abordagens promissoras para superar essa limitação são realizar a regressão por Processos Gaussianos (GP) para modelar as incertezas e utilizar técnicas de otimização Bayesiana. Visando iniciar uma maior discussão dos problemas numéricos esperados com a aplicação destas novas abordagens, o objetivo deste trabalho é estudar o efeito do otimizador no desempenho de sistemas RTO inicializados aleatoriamente e na presença ruídos de medição. Para isto, são consideradas as técnicas MA com adaptação via GPs (MA-GP) e uma nova abordagem por otimização Bayesiana restrita através da função de aquisição *Expected Improvement with Constraints* (EIC). A partir da convergência ao ótimo da planta, viabilidade das iterações e do custo computacional, compara-se o desempenho de algoritmos de otimização não linear determinísticos (*Sequential Quadratic Programming* (SQP) e *Nelder-Mead Simplex* (NM)) e uma heurística evolutiva (*Differential Evolution* (DE)). Para dois modelos de *benchmark* disponíveis na literatura, ilustra-se que os algoritmos SQP e NM podem falhar em encontrar o ótimo global durante as iterações do sistema RTO. Como resultado é observada uma queda no desempenho do sistema, o qual apresenta uma maior variabilidade e sensibilidade à etapa de inicialização. Para um intervalo de confiança de 95% o DE superou os demais algoritmos, embora necessite de um esforço computacional mais elevado. Além disso, é possível comprovar o potencial da nova técnica proposta via otimização Bayesiana restrita. Ao permitir o uso do algoritmo irrestrito NM, ela se torna eficiente quando comparada ao MA-GP com SQP, demandando ao mesmo tempo um baixo custo computacional. Entretanto, sua convergência ainda precisa de uma maior investigação e seu desempenho é inferior ao MA-GP para problemas onde o ponto de operação ótimo da planta está na interseção de suas restrições.

**Palavras-chave:** Otimização em tempo-real. Modifier Adaptation. Processos Gaussianos. Otimização Bayesiana. Otimização não linear.



# Abstract

Real-time optimization (RTO) is a technique capable of iteratively leading an industrial process (plant) to its optimal economic operation, using for this an approximate mathematical model combined with the solution of a non-linear optimization problem. In order to deal with model-plant mismatch, different approaches are available in the literature, among which the Modifier Adaptation (MA) stands out. It applies first-order corrections to the cost and constraints functions in order to achieve plant optimality upon convergence. However, the calculation of these corrections depends on plant gradient information, which is difficult to obtain. Promising approaches to overcome this limitation are to perform Gaussian Processes (GP) regression to model the mismatch and use Bayesian optimization techniques. Aiming to initiate a further discussion of the numerical problems expected with the application of these new approaches, the objective of this work is to study the effect of the optimizer on the performance of randomly initialized RTO systems in the presence of measurement noise. For this, we consider the MA with GPs adaptation (MA-GP) and a new approach using constrained Bayesian optimization through the Expected Improvement with Constraints (EIC) acquisition function. Based on the convergence to the plant optimum, iteration feasibility and computational cost, the performance of deterministic nonlinear optimization algorithms (Sequential Quadratic Programming (SQP) and Nelder-Mead Simplex (NM)) and an evolutionary heuristic (Differential Evolution (DE)) are compared. For two benchmark models available in the literature, it is illustrated that the SQP and NM algorithms may fail to find the optimum during the RTO system iterations. As a result, the system's performance is degraded, presenting higher variability and sensitivity to the initialization step. For a confidence interval of 95%, DE outperformed the other algorithms, although it requires a higher computational effort. Furthermore, it is possible to prove the proposed technique's potential via constrained Bayesian optimization. By allowing the use of the unrestricted NM algorithm, it becomes efficient when compared to MA-GP with SQP, requiring at the same time a low computational cost. However, its convergence is still uncertain and its performance is inferior to MA-GP, especially for problems where the plant's optimal operating point is at the intersection of its constraints.

**Keywords:** Real-time Optimization. Modifier Adaptation. Gaussian Processes. Bayesian Optimization. Nonlinear Optimization.

# Lista de ilustrações

Figura 1 – Hierarquia de decisão em uma planta . . . . .	15
Figura 2 – Elementos de um sistema de otimização em tempo-real . . . . .	16
Figura 3 – Representação gráfica do princípio por trás da adição de modificadores . . . . .	25
Figura 4 – Exemplo de predições com processos Gaussianos . . . . .	35
Figura 5 – Exemplo de otimização Bayesiana com a função de aquisição EI . . . . .	37
Figura 6 – Diferença relativa da função objetivo para as execuções de validação . . . . .	52
Figura 7 – Superfície de decisão do MA-GP-EIC com SQP para o RWO . . . . .	53
Figura 8 – Diferença relativa da função objetivo para o MA-GP . . . . .	54
Figura 9 – <i>Boxplot</i> do custo computacional das iterações do MA-GP . . . . .	55
Figura 10 – Diferença relativa da função objetivo para o MA-GP-EIC . . . . .	55
Figura 11 – <i>Boxplot</i> do custo computacional das iterações do MA-GP-EIC . . . . .	56
Figura 12 – Diferença relativa dos sinais de controle para o MA-GP-EIC . . . . .	56
Figura 13 – Restrições do MA-GP-EIC para o problema RWO . . . . .	57
Figura 14 – Diferença relativa da função objetivo para os melhores otimizadores. . . . .	57
Figura 15 – Diferença relativa dos sinais de controle para os melhores otimizadores. . . . .	58
Figura 16 – Restrições para o problema RAP . . . . .	58
Figura 17 – Restrições para o problema RWO . . . . .	59
Figura 18 – <i>Boxplot</i> do custo computacional do MA-GP com SQP e MA-GP-EIC com NM . . . . .	60
Figura 19 – Diferença relativa da função objetivo para os otimizadores SQP e NM. . . . .	61

# Lista de tabelas

Tabela 1 – Valores dos parâmetros e condições iniciais para o RAP . . . . .	49
---	----

# Lista de abreviaturas e siglas

RTO	Otimização em Tempo-Real
MA	<i>Modifier Adaptation</i>
GP	Processos Gaussianos
MA-GP	<i>Modifier Adaptation</i> com Processos Gaussianos
EI	<i>Expected Improvement</i>
EIC	<i>Expected Improvement with Constraints</i>
MA-GP-EIC	<i>Modifier Adaptation</i> com Função de Aquisição EIC
OB	Otimização Bayesiana
SQP	Algoritmo <i>Sequential Quadratic Programming</i>
DE	Algoritmo <i>Differential Evolution</i>
NM	Algoritmo <i>Nelder-Mead Simplex</i>
IC	Intervalo de confiança

# Lista de símbolos

$\Phi$	Função de distribuição cumulativa
$\phi$	Função densidade de probabilidade

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	Justificativa	18
1.2	Objetivos	19
1.3	Organização	19
1.4	Contribuições	20
1.4.1	Produção Bibliográfica	20
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>21</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>30</b>
3.1	Introdução	30
3.2	Otimização não linear	30
3.2.1	O algoritmo Sequential Quadratic Programming	32
3.2.2	O algoritmo Differential Evolution	32
3.2.3	O algoritmo Nelder-Mead Simplex	33
3.3	Processos Gaussianos	34
3.4	Otimização Bayesiana	35
<b>4</b>	<b>METODOLOGIA</b>	<b>39</b>
4.1	Introdução	39
4.2	Modifier Adaptation com Processos Gaussianos	39
4.3	Modifier Adaptation com função de aquisição EIC	40
4.4	Desenho do Experimento	42
4.4.1	Comparação dos intervalos de confiança	43
4.5	Etapas e desafios da implementação	45
4.5.1	Modelos de processo	45
4.5.2	Otimizadores	46
4.5.3	Processos Gaussianos	46
4.5.4	Sistema RTO	46
4.5.5	Experimento	46
4.6	Modelos de Processo	47
4.6.1	O Reator de Williams-Otto (RWO)	47
4.6.2	O Reator de Acetoacilação de Pirrol (RAP)	49
<b>5</b>	<b>RESULTADOS</b>	<b>51</b>
5.1	Introdução	51

<b>5.2</b>	<b>Recursos computacionais</b> . . . . .	<b>51</b>
<b>5.3</b>	<b>Dados Experimentais</b> . . . . .	<b>51</b>
<b>5.4</b>	<b>Validação da implementação</b> . . . . .	<b>52</b>
<b>5.5</b>	<b>Escolha do algoritmo de otimização</b> . . . . .	<b>53</b>
5.5.1	MA-GP . . . . .	54
5.5.2	MA-GP-EIC . . . . .	54
<b>5.6</b>	<b>Comparação entre RTO MA-GP e MA-GP-EIC</b> . . . . .	<b>56</b>
<b>6</b>	<b>CONCLUSÃO</b> . . . . .	<b>62</b>
<b>6.1</b>	<b>Considerações Finais</b> . . . . .	<b>62</b>
<b>6.2</b>	<b>Propostas de Continuidade</b> . . . . .	<b>63</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>65</b>

# 1 Introdução

A Indústria 4.0 deixou de ser apenas um conceito e já é uma realidade, sendo progressivamente mais adotada pelas principais empresas do setor industrial. Sua adoção provoca uma série de mudanças no processo produtivo, a qual torna possível a coleta e integração de dados de fontes distintas, tendo como objetivo final atingir maiores níveis de eficiência operacional, produtividade e automatização (LU, 2017). Com a crescente disponibilidade de dados, internet das coisas (IoT) e computação em nuvem (LASI et al., 2014), o advento de abordagens orientadas a dados para otimização de processos industriais vem ganhando mais visibilidade e aplicações (GOPALUNI et al., 2020).

A hierarquia de controle e tomada de decisão em uma planta industrial pode ser dividida em cinco níveis (DARBY et al., 2011), como pode ser visto na Figura 1. Cada nível tem como objetivo otimizar a produção a partir de previsões e modelos do processo, porém com diferentes impactos, incertezas e frequência. No nível de planejamento, os principais objetivos são definir o que produzir e de qual forma. Isso engloba, por exemplo, quais insumos comprar ou produzir e em qual quantidade. O nível de programação atua em como as ações e eventos de um plano escolhido devem ser executadas ao longo do tempo, definindo, por exemplo, a alocação de recursos e a logística de movimentação de materiais. A tomada de decisão é geralmente feita a partir da resolução de problemas de otimização linear inteira mistos (BIEGLER; ZAVALA, 2009) nestes níveis.

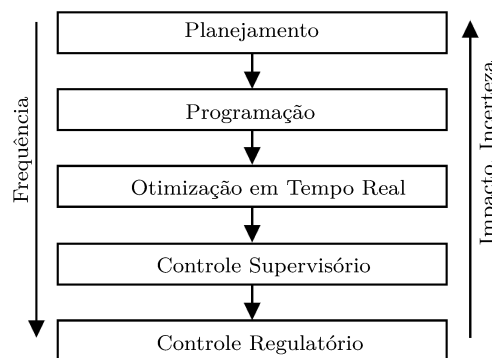


Figura 1 – Hierarquia de decisão em uma planta. A direção das setas indicam o aumento da grandeza em questão

O nível de otimização em tempo-real (RTO, do inglês *Real-Time Optimization*) é o primeiro que interage efetivamente com o processo. Ele é responsável por definir valores de referência de acordo com um objetivo econômico, os quais serão posteriormente utilizados pelos níveis de controle (ZHANG; MONDER; FORBES, 2002). No nível de controle supervisório é feito o controle dinâmico do processo, onde ainda é possível a aplicação de técnicas de otimização para esta finalidade (DARBY et al., 2011). Uma abordagem onde RTO e controle supervisório são integrados em um único nível também é possível,



conforme analisado em (ELLIS; DURAND; CHRISTOFIDES, 2014). O nível mais baixo é o de controle regulatório, tipicamente executado por controladores PID e em uma escala de tempo abaixo de segundos, atuando diretamente no processo (DARBY et al., 2011).

Os elementos que compõem um sistema RTO típico são exibidos na Figura 2. Uma premissa para o seu funcionamento é a existência de um rigoroso modelo do processo. Portanto, o primeiro passo do projeto deste sistema consiste na coleta de dados para sua definição. Em seguida, é formulado um problema de otimização baseado neste modelo, que calcula as entradas ótimas em relação a um objetivo econômico, a partir de diferentes premissas operacionais, como preço de insumos, restrições de inventário e qualidade do produto final. Como simplificação, assume-se que exista um sistema de controle capaz de aplicar as referências calculadas pelo otimizador. A partir desta definição inicial, periodicamente o sistema irá coletar novos dados do processo e automaticamente utilizá-los na estratégia de adaptação escolhida. É importante ressaltar que os novos dados coletados precisam passar por um rigoroso processo de validação, tipicamente na forma de verificação de limites aceitáveis, ou através de técnicas estatísticas de reconciliação e detecção de erros grosseiros (BAGAJEWICZ, 2000).

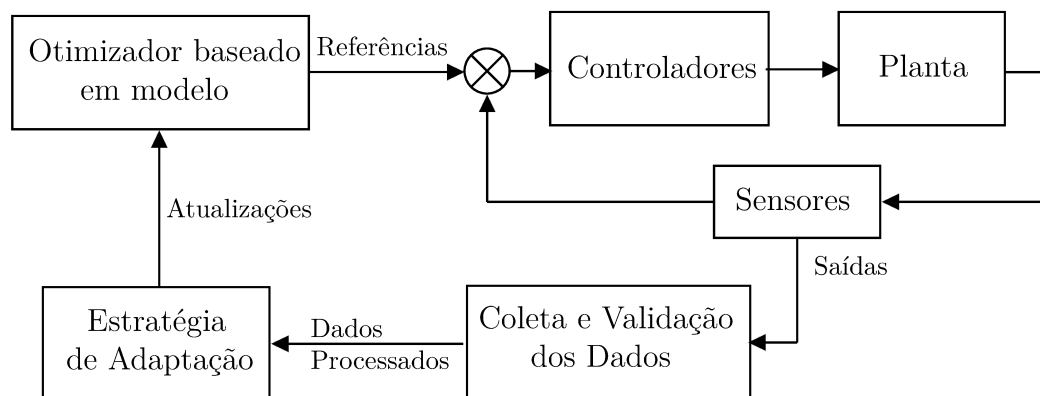


Figura 2 – Elementos de um sistema de otimização em tempo-real

Na prática, a definição do modelo é uma tarefa complexa e que demanda bastante tempo, o que resulta na introdução de incertezas que podem causar situações de operação sub-ótima ou ainda ineficazes quando existem restrições (MARCHETTI; CHACHUAT; BONVIN, 2009). O uso de estratégias de adaptação do modelo é essencial para lidar com as discrepância modelo-planta e tentar buscar a convergência do sistema, estando disponíveis diversas abordagens na literatura. Sistemas RTO podem ser classificados de acordo com a forma de adaptação realizada a partir das amostras. Neste caso, existem três possibilidades (MARCHETTI et al., 2016):

- a) **Adaptação de parâmetros do modelo:** é a abordagem mais intuitiva, sendo conhecida na literatura como método *two-step* (TS). Em cada iteração, amostras das saídas do processo são coletadas, processadas e posteriormente utilizadas para refinar o modelo do processo. Em seguida, o modelo atualizado é usado

na otimização dos valores de referência do processo de acordo com a função objetivo e restrições estabelecidas (NAYSMITH; DOUGLAS, 1995; MARLIN; HRYMAK et al., 1997; YIP; MARLIN, 2002; FORBES; MARLIN, 1996).

- b) **Adaptação de modificadores:** ao contrário do método TS, nesta abordagem o modelo do processo é fixo. Em cada iteração, a adaptação é realizada diretamente na função objetivo e restrições do problema de otimização. Isso é feito através da adição de termos modificantes que representam as discrepâncias entre a planta e o modelo, calculados a partir das medições. Uma particularidade deste método consiste na necessidade de se medir gradientes da planta em relação às entradas, ao invés de somente as suas saídas, o que na prática pode ser desafiador. É conhecido na literatura como *Modifier Adaptation* (MA) (CHACHUAT; SRINIVASAN; BONVIN, 2009; MARCHETTI; CHACHUAT; BONVIN, 2010).
- c) **Adaptação direta de entradas:** diferentemente das anteriores, esta abordagem não utiliza técnicas de otimização numérica. Ela transforma o problema de otimização em um de controle em malha-fechada capaz de calcular as entradas do sistema (FRANÇOIS; SRINIVASAN; BONVIN, 2005). Seu objetivo é encontrar funções das variáveis medidas que quando mantidas constantes pelo ajuste das entradas do processo, garantem levar a planta ao ótimo. Os métodos mais comuns dessa abordagem são o *Extreme Seeking Control* (KRSTIĆ; WANG, 2000), *Neighboring-Extremal Control* (GROS; SRINIVASAN; BONVIN, 2009) e *Self-Optimizing Control* (SKOGESTAD, 2000)

Os métodos anteriores são executados de forma iterativa, na esperança de se atingir o ótimo da planta após algumas iterações. O método a) tem a garantia de convergência quando as condições de adequação do modelo (FORBES; MARLIN; MACGREGOR, 1994) são respeitadas, o que infelizmente é algo difícil de se atingir e verificar. O método b) em geral possui sua convergência garantida mesmo com discrepâncias modelo-planta estruturais (MARCHETTI et al., 2016), enquanto as técnicas de c) impõe o respeito às condições necessárias em suas formulações. Uma discussão sobre a convergência destes métodos em diversos cenários pode ser encontrada em (SRINIVASAN; BONVIN, 2019).

De acordo com (MARCHETTI et al., 2020), aplicações de RTO são majoritariamente estudadas utilizando processos químicos. Neste contexto, uma iteração do RTO é geralmente feita após a detecção de que a planta esteja em estado estacionário. Isso é válido para processos que operam de forma contínua, porém para processos descontínuos cada iteração é realizada ao final de uma batelada, tendo como objetivo determinar perfis ótimos para as entradas ao invés de somente valores de referência. A otimização desses perfis é um problema de otimização em tempo-real dinâmica (DRTO), o qual geralmente é reformulado como um problema estático (COSTELLO et al., 2011).

## 1.1 Justificativa

Com o avanço da indústria 4.0, o uso de abordagens orientadas à dados, em especial técnicas de aprendizado de máquina, passou a ser um elemento fundamental na otimização de processos (CHIANG; LU; CASTILLO, 2017). A integração do RTO com essas técnicas é uma proposta muito interessante, e tem bastante potencial para auxiliar na solução das principais dificuldades enfrentadas pelas abordagens clássicas, embora ainda pouco explorada na literatura.

Recentemente, (FERREIRA et al., 2018) utilizou regressão por Processos Gaussianos como forma de contornar a necessidade de estimativa de gradientes para a atualização dos modificadores. O uso de técnicas de otimização por região de confiança (GAO; WENZEL; ENGELL, 2016) em conjunto com essa abordagem é explorado em (CHANONA et al., 2019). Em (ZHANG et al., 2019), o autor estuda a viabilidade do uso de modelos híbridos (ZENDEHBOUDI; REZAEI; LOHI, 2018) em um sistema RTO, através de uma rede-neural em conjunto com um modelo de processo baseado em princípios fundamentais. Outra aplicação interessante consiste no uso de aprendizado por reforço à um problema DRTO com incertezas (PETSAGKOURAKIS et al., 2020).

Isso mostra o crescente interesse e potencial para o desenvolvimento de novas aplicações e estratégias para a integração do RTO com abordagens orientadas à dados. Seguindo esta linha, (RIO-CHANONA et al., 2020) foi o primeiro a propor e aplicar com sucesso a integração de conceitos de otimização Bayesiana irrestrita, abrindo um novo horizonte de possibilidades para a solução deste problema. Esta abordagem se mostrou bastante promissora, o que motiva a investigação de métodos Bayesianos restritos para um potencial aumento de sua eficiência.

O RTO é um problema de otimização não linear aplicado a processos industriais, cujo principal desafio é levar a planta ao seu ponto de operação ótimo iterativamente a partir de um modelo aproximado do processo. Um elemento essencial deste sistema é a solução do problema de otimização baseado em modelos, geralmente não linear. Entretanto, pouca atenção é dada na literatura aos problemas numéricos desta etapa, apesar dos impactos negativos no desempenho do sistema RTO serem conhecidos (QUELHAS; JESUS; PINTO, 2013).

A adoção de técnicas de aprendizado de máquina pode resultar na acentuação destes efeitos indesejados. Portanto, fatores como a presença de ruídos de medição, dados de treinamento iniciais e escolha do algoritmo de otimização não devem ser negligenciados neste contexto. Além disso, o desenho de experimentos adequados para a comparação estatística de desempenho de abordagens RTO se torna necessário. Principalmente para trabalhos mais antigos, observa-se a falta de uma metodologia adequada na literatura para essa análise, como as apresentadas em (ANDERSSON; BRADFORD; IMSLAND, 2020;

RIO-CHANONA et al., 2020).

## 1.2 Objetivos

O objetivo geral deste trabalho é estudar o efeito do otimizador no desempenho de sistemas RTO inicializados com pontos de operação factíveis aleatórios e contaminados por ruídos de medição Gaussiano. Para isto, são consideradas as técnicas MA com adaptação via Processos Gaussianos (FERREIRA et al., 2018) e uma proposta de extensão de (RIO-CHANONA et al., 2020) sob a ótica da otimização Bayesiana restrita através da função de aquisição *Expected Improvement with Constraints* (EIC) (GELBART, 2015). A partir da convergência ao ótimo da planta, viabilidade das iterações e do custo computacional, é possível comparar o desempenho de algoritmos de otimização não linear determinísticos e uma heurística evolutiva. Isso permitirá averiguar o quão adequados são os algoritmos para cada modelo de processo e posteriormente a comparação de desempenho entre os sistemas RTO considerados.

A fim de alcançar o objetivo geral apresentado, definiu-se os seguintes objetivos específicos:

- Implementar dois modelos de processo de *benchmark* com diferentes níveis de complexidade
- Implementar otimizador baseado em modelos para os algoritmos SQP, *Nelder-Mead Simplex* e *Differential Evolution*
- Implementar o algoritmo MA com Processos Gaussianos (MA-GP)
- Desenvolver o algoritmo MA com Processos Gaussianos e função de aquisição EIC (MA-GP-EIC)
- Validar a implementação dos sistemas RTO sobre os modelos de *benchmark*
- Definir o otimizador mais adequado para o MA-GP e MA-GP-EIC
- Comparar o desempenho entre MA-GP e MA-GP-EIC para os otimizadores mais adequados

## 1.3 Organização

No Capítulo 2 é apresentada uma revisão da literatura relacionada a sistemas RTO, onde conta-se a evolução da técnica ao longo dos anos e suas principais variações. No Capítulo 3 é feita uma breve descrição dos fundamentos teóricos relacionados ao problema, bem como uma maior detalhamento das técnicas utilizadas para a sua solução neste trabalho. No Capítulo 4 é apresentada a metodologia, contendo a descrição dos modelos

considerados para os estudos de caso, desenho dos experimentos, e descrição das técnicas RTO avaliadas. O Capítulo 5 apresenta a análise para os resultados dos experimentos de cada objetivo específico. Finalmente, no Capítulo 6 são mostradas as conclusões e sugestões para continuidade deste trabalho.

## 1.4 Contribuições

Este trabalho possui duas contribuições principais. A primeira consiste em estudar uma deficiência observada na literatura, que é o impacto da escolha do algoritmo de otimização no desempenho dos sistemas RTO via MA-GP. A segunda consiste na proposta de uma nova abordagem que reformula o problema RTO sob a ótica da otimização Bayesiana restrita, utilizando a função de aquisição EIC. Um contribuição secundária foi a consolidação da implementação no formato de um pacote<sup>1</sup> *Python*, facilitando a reprodução dos resultados e extensão deste trabalho por outros pesquisadores

### 1.4.1 Produção Bibliográfica

Parte dos resultados que serão apresentados no Capítulo 5 foram publicados na *5<sup>th</sup> IEEE Conference on Control Technology and Applications (CCTA 2021)*.

- RUELA, V. S. P.; BESSANI, M. Investigation of Initial Data and Optimizer in Real-Time Optimization Performance Via Modifier Adaptation with Gaussian Processes. *5<sup>th</sup> IEEE Conference on Control Technology and Applications. San Diego - CA. 2021.*

---

<sup>1</sup> <<https://github.com/vicrsp/rto>>

## 2 Revisão da literatura

O problema de otimizar o desempenho em estado estacionário de uma determinada planta industrial sujeito a restrições operacionais e de segurança pode ser formulado como o seguinte problema de otimização não linear (CHACHUAT; SRINIVASAN; BONVIN, 2009):

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimizar}} \quad & G_0^p(\mathbf{u}) := g_0(\mathbf{u}, \mathbf{y}^p(\mathbf{u})) \\ \text{s.t.} \quad & G_i^p(\mathbf{u}) := g_i(\mathbf{u}, \mathbf{y}^p(\mathbf{u})) \leq 0, \quad i = 1, \dots, n_g \\ & \mathbf{u} \in \mathcal{U} \end{aligned} \quad (2.1)$$

onde  $g_0 \in \mathbb{R}$  é um indicador de desempenho da planta a ser minimizado,  $g_i : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ ,  $i = 1, \dots, n_g$ , são as restrições de desigualdade,  $\mathbf{y}^p \in \mathbb{R}^{n_y}$  as saídas medidas do processo e  $\mathbf{u} \in \mathbb{R}^{n_u}$  representa o conjunto de variáveis de entrada, ou controláveis;  $\mathcal{U}$  representa o espaço de busca das variáveis de decisão, o qual é tipicamente definido como limites superiores e inferiores,  $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^{n_u} : \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U\}$ . O sobrescrito  $(\cdot)^p$  indica uma quantidade referente à planta.

Note que para a solução deste problema será necessário excitar a planta com um número elevado de entradas, o que pode se tornar proibitivo do ponto de vista prático, especialmente se o algoritmo de otimização escolhido necessitar do cálculo de gradientes. Portanto, uma prática comum é desenvolver um modelo matemático que aproxime da melhor forma possível a resposta da planta, definindo um novo problema de otimização estático:

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimizar}} \quad & G_0(\mathbf{u}) := g_0(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \\ \text{s.t.} \quad & G_i(\mathbf{u}) := g_i(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \leq 0, \quad i = 1, \dots, n_g \\ & \mathbf{u} \in \mathcal{U} \end{aligned} \quad (2.2)$$

onde  $\boldsymbol{\theta}$  é o conjunto de parâmetros do modelo.

A complexidade deste problema está diretamente relacionada ao modelo de processo utilizado, o qual por serem geralmente descritos por sistemas de equações diferenciais podem introduzir não linearidades na função objetivo e restrições. Para processos químicos contínuos operando em estado estacionário, as variáveis de decisão são simplesmente valores escalares (*setpoints*). Por outro lado, para processos que operam em forma descontínua (batelada) é necessário determinar perfis de controle variantes no tempo, o que caracteriza um problema de otimização dinâmica. A parametrização de cada variável em intervalos pré-definidos permite transformá-lo no problema estático descrito pela Equação 2.2 (CHACHUAT; SRINIVASAN; BONVIN, 2009). Note que isso pode aumentar consideravelmente a quantidade de variáveis de decisão e, conseqüentemente, o esforço computacional para sua solução.

Outro fator a se considerar é o nível de detalhe empregado na definição da estrutura do modelo do processo. A modelagem é, em sua maioria, baseada em princípios fundamentais e aproximações de fenômenos complexos, o que torna factível a solução deste problema utilizando algoritmos determinísticos (BIEGLER, 1992). Entretanto, mesmo com os recentes avanços de poder computacional, a introdução de modelos detalhados de simulação de fenômenos complexos, como dinâmica de fluidos e transferência de calor, ainda é proibitivo do ponto de vista de otimização (SAVAGE et al., 2021).

Isso faz com que na prática somente modelos aproximados do processo sejam utilizados, o que introduz discrepâncias estruturais entre a planta e o modelo na maioria das aplicações. Dessa forma, a solução ótima do Problema 2.2 poderá não ser equivalente à do Problema 2.1. Para lidar com este tipo de incerteza, foram desenvolvidos os chamados sistemas de otimização em tempo-real (RTO), os quais introduzem estratégias de adaptação em suas formulações para tratar, mas não eliminar, as incertezas entre planta e modelo. De acordo com (FRANÇOIS; SRINIVASAN; BONVIN, 2012), sistemas RTO podem ser classificadas entre as abordagens implícita ou explícita.

Métodos implícitos também são conhecidos na literatura como adaptação direta de entradas. O conceito por trás deles consiste em selecionar saídas cujos valores ótimos são aproximadamente invariantes à incertezas, e então tentar mantê-las constantes com o ajuste de sinais de controle. Os principais métodos presentes na literatura são o *Necessary Conditions of Optimality Tracking* (SRINIVASAN; BONVIN, 2007), *Neighboring-Extremal Control* (GROS; SRINIVASAN; BONVIN, 2009), *Extreme Seeking Control* (KRSTIĆ; WANG, 2000) e o *Self-Optimizing Control* (SKOGESTAD, 2000). Métodos implícitos não serão abordados neste trabalho, sendo recomendada a leitura das referências anteriores para um maior detalhamento.

Por outro lado, métodos explícitos utilizam otimização numérica para lidar com as incertezas e adaptar, de acordo com medições, parâmetros de modelo ou fatores de correção no problema estático antes de utilizá-los para o cálculo dos sinais de controle. Dentro desta classe, sistemas RTO baseados em ajuste de parâmetros do modelo são os que possuem o maior número de aplicações reportadas na literatura (DARBY et al., 2011; CÂMARA; QUELHAS; PINTO, 2016), uma vez que é uma abordagem intuitiva e requer somente amostras das saídas do processo, cuja medição é mais fácil na prática. O método mais comum é o por identificação e otimização repetitiva, conhecido também na literatura como *two-step* (TS), o qual foi inicialmente proposto por (JANG; JOSEPH; MUKAI, 1987).

Em cada iteração, o método TS utiliza como estratégia de adaptação o ajuste de um sub-conjunto de parâmetros do modelo a partir de dados coletados do processo, definindo um problema de reconciliação de dados (NARASIMHAN; JORDACHE, 1999). Sua função objetivo pode ser definida de forma determinística usando técnicas clássicas de

regressão, como o estimador de mínimos quadrados ponderados (BARD, 1974):

$$\underset{\boldsymbol{\theta}_{adj}}{\text{minimizar}} \quad \sum_{i=1}^{n_E} \sum_{j=1}^{n_y} \frac{[\mathbf{y}^P(\mathbf{u})_{i,j} - \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_{adj})_{i,j}]^2}{\sigma_{i,j}^2} \quad (2.3)$$

onde  $n_E$  é o número de experimentos,  $\sigma_{i,j}^2$  é a variância da variável dependente de índice  $j$  no experimento  $i$ , e  $\boldsymbol{\theta}_{adj} \subseteq \boldsymbol{\theta}$  representa os parâmetros ajustáveis do modelo aproximado do processo. É importante ressaltar que restrições relativas ao processo podem ser adicionadas ao problema anterior para evitar a escolha de parâmetros que levem a soluções infactíveis.

O problema de identificação de parâmetros pode se tornar bastante complexo devido ao grande número de parâmetros a serem ajustados e da função objetivo apresentar uma natureza multimodal (SCHWAAB et al., 2008). A escolha do algoritmo de otimização para a sua solução não é uma tarefa simples, sendo bastante estudada na literatura. Em (CÂMARA et al., 2017), o autor realiza uma extensa análise dos aspectos numéricos deste problema, além de disponibilizar uma compilação com diferentes aplicações que reportam o uso de dados reais. É possível constatar o emprego de uma variedade de algoritmos de otimização determinísticos e estocásticos, evidenciando a necessidade de um grande esforço somente na solução desta etapa da abordagem TS.

Em sistemas RTO, a solução deste problema deve ser feita de forma automática, o que necessita de atenção em relação à qualidade dos dados utilizados para o seu ajuste. O processo de medição deve garantir que haja informação suficiente nas amostras coletadas, o que pode ser atingido através de técnicas de planejamento estatístico de experimentos (WEISSMAN; ANDERSON, 2015), por exemplo. Além disso, novos dados coletados precisam passar por uma série de validações e técnicas de detecção de erros grosseiros (BAGAJEWICZ, 2000). De fato, a interação entre as etapas de identificação e otimização estática é bem crítica, pois o ajuste dos parâmetros do modelo pode levar ao cálculo de um ponto de operação ainda mais distante do ótimo da planta, uma vez que minimizar a soma dos erros quadráticos pode não estar relacionado com a factibilidade e otimalidade da planta, dada a discrepância planta-modelo existente (CHACHUAT; SRINIVASAN; BONVIN, 2009).

Se o modelo de processo respeita os critérios de adequação estabelecidos por (FORBES; MARLIN; MACGREGOR, 1994) o método TS convergirá para o ponto ótimo de operação da planta. Um modelo é dito adequado se existe um conjunto de parâmetros cuja solução ótima é equivalente à da planta. Matematicamente, isso é verificado através respeito das restrições do modelo, gradientes nulos e Hessiana positiva definida para a objetivo econômico do RTO e do problema de identificação baseados no modelo (FORBES; MARLIN; MACGREGOR, 1994). Respeitar estas condições de otimalidade é muito difícil na prática, principalmente pois requer o conhecimento do ótimo da planta, o qual é geralmente desconhecido. Portanto, novas abordagens foram desenvolvidas visando aprimorar a convergência do método TS, uma vez que a possibilidade de ganhos com o uso de sistemas



RTO são muito grandes (DARBY et al., 2011).

De forma geral, as demais abordagens consistem em modificar a função objetivo e restrições do Problema 2.2 de modo que as condições de KKT da planta e modelo possam se tornar equivalentes, correspondendo à estratégia de adaptação de modificadores. Seja  $\mathbf{u}_k$  o ponto de operação atual, o problema estático com modificadores para determinar a próxima iteração do RTO é formulado como (CHACHUAT; SRINIVASAN; BONVIN, 2009):

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimizar:}} \quad & G_0(\mathbf{u}) + \epsilon_k^{G_0} + (\lambda_k^{G_0})^T (\mathbf{u} - \mathbf{u}_k) \\ \text{sujeito a:} \quad & G_i(\mathbf{u}) + \epsilon_k^{G_i} + (\lambda_k^{G_i})^T (\mathbf{u} - \mathbf{u}_k) \leq 0, \quad \forall i = 1, \dots, n_g \\ & \mathbf{u} \in \mathcal{U} \end{aligned} \quad (2.4)$$

onde  $\epsilon_k^{G_i} \in \mathbb{R}$ ,  $i = 0, \dots, n_g$  e  $\lambda_k^{G_i} \in \mathbb{R}^{n_u}$ ,  $i = 0, \dots, n_g$  são os termos modificadores da  $k$ -ésima iteração, definidos como:

$$\epsilon_k^{G_i} = G_i^p(\mathbf{u}_k) - G_i(\mathbf{u}_k), \quad i = 0, \dots, n_g \quad (2.5)$$

$$(\lambda_k^{G_i})^T = \frac{\partial G_i^p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial G_i}{\partial \mathbf{u}}(\mathbf{u}_k), \quad i = 0, \dots, n_g \quad (2.6)$$

Os termos  $\epsilon_k^{G_i}$  correspondem ao viés que representa a diferença entre as saídas da planta e os previstos pelo modelo, enquanto  $\lambda_k^{G_i}$  representam a diferença entre os gradientes reais e previstos. Como o viés da função objetivo é uma constante, ele é geralmente omitido em formulações disponíveis na literatura pois não afeta o processo de otimização. A forma como estes modificadores são utilizados caracteriza as diferentes abordagens presentes na literatura. O efeito da adição de modificadores pode ser observado graficamente na Figura 3. É interessante notar que se não existir discrepâncias entre a planta e o modelo, os modificadores serão nulos e não será necessário nenhum ajuste. Observe que o objetivo é tentar igualar o gradiente da planta e modelo para que as condições KKT ocorram no mesmo ponto de operação. No método TS, isso poderia ser atingido somente com a existência de um modelo adequado, mostrando o potencial desta abordagem para relaxar os critérios estabelecidos por (FORBES; MARLIN, 1994). Entretanto, ela introduz a necessidade de estimativa dos gradientes da planta em relação ao ponto de operação atual, a qual é bastante desafiadora na prática.

Uma das primeiras técnicas usando modificação de adaptadores foi proposta por (ROBERTS, 1979), sendo conhecido na literatura como ISOPE (*Integrated System Optimization and Parameter Estimation*). Ele é formulado como uma extensão do método TS, caracterizado pela adição de somente um modificador na função objetivo com o intuito de introduzir uma melhor integração entre os problemas de otimização estático e de identificação. Os esforços relacionados a esta técnica concentraram-se em melhorar a

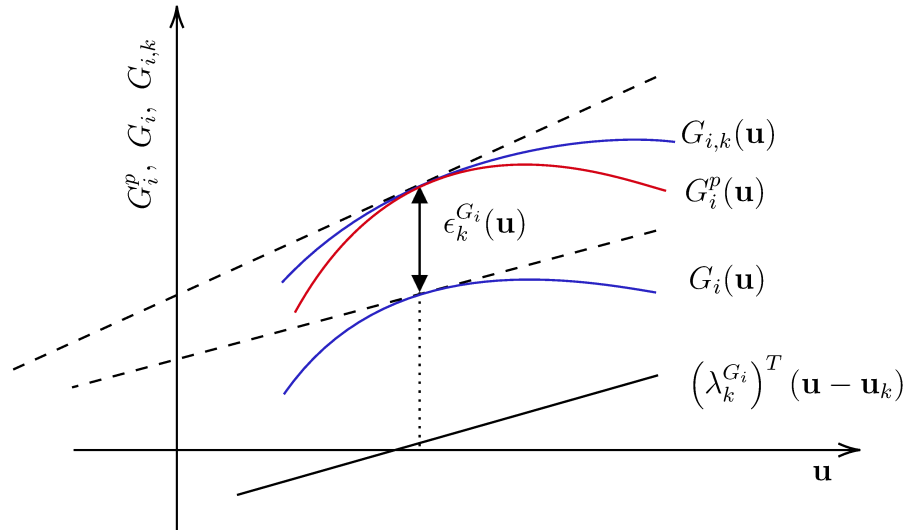


Figura 3 – Representação gráfica do princípio por trás da adição de modificadores para uma restrição  $G_i$ . Em cada iteração, aplica-se os termos modificadores de forma que após  $k$  iterações a solução do modelo e planta sejam equivalentes. Adaptada de (MARCHETTI et al., 2016)

robustez da estimativa dos gradientes (ROBERTS, 1995), a qual teve um grande avanço após o desenvolvimento do método ISOPE Dual (BRDYŚ; TATJEWSKI, 1995). Esta nova técnica introduz a propriedade de dualidade, no qual o sinal de controle atual é gerado não só para satisfazer o objetivo de otimização, mas também fornecer informações suficientes para a etapa de identificação futura. Em seguida, (TATJEWSKI, 2002) mostra que a convergência do ISOPE para o ótimo real não depende do problema de identificação, mas somente da diferença entre o gradiente do processo e da planta. Esse resultado é considerado um marco para as técnicas por modificação de adaptadores, uma vez que é comprovada a possibilidade de relaxar as premissas de adequação do modelo estabelecidas por (FORBES; MARLIN; MACGREGOR, 1994).

Uma abordagem alternativa ao ISOPE é considerar somente o viés das restrições como forma de adaptação, que são os métodos conhecidos na literatura como *Constraint Adaptation* (CA). Proposto inicialmente por (FORBES; MARLIN, 1994), ele surgiu da observação de que o potencial de otimização vem da escolha correta de quais restrições devem estar ativas. Em (CHACHUAT; MARCHETTI; BONVIN, 2008), o autor apresenta uma discussão sobre o método, como detectar este conjunto ativo e sua convergência. Ele mostra que quando o ponto de operação ótimo está nas restrições do problema estático, ganhos de desempenho são obtidos em poucas iterações e sem a necessidade de estimar gradientes da planta. Em contrapartida, ele pode convergir para um ponto de operação sub-ótimo se o modelo do processo não possuir qualidade adequada (FORBES; MARLIN; MACGREGOR, 1994). Como não depende de gradientes da planta, esta abordagem possui potencial para ser mais facilmente escalável para sistemas maiores e mais complexos.

Diferente do ISOPE, que considera somente os modificadores na função objetivo, e

do CA que modifica somente as restrições, (MARCHETTI; CHACHUAT; BONVIN, 2009) desenvolveu uma nova metodologia que considera as duas ao mesmo tempo, generalizando a etapa de adaptação dos modificadores descrita pelo Problema 2.4. A partir dos resultados de (TATJEWSKI, 2002), os autores mostram que utilizando estes modificadores as condições de adequação do modelo podem ser consideravelmente relaxadas, o que é uma grande vantagem. Este método é conhecido na literatura como *Modifier Adaptation* (MA). O MA permite que o sistema RTO atinja um ponto que atende as condições de KKT da planta após sua convergência (MARCHETTI et al., 2016), porém ao custo da necessidade de estimativa dos gradientes.

Na prática, estimá-los ainda é um dos maiores gargalos desta abordagem, uma vez que sua medição direta é difícil e ruídos estarão sempre presentes, principalmente em ambientes industriais (MARCHETTI et al., 2016). O problema de estimativa de gradientes é recorrente na literatura, sendo fácil constatar que a evolução desta abordagem em muitos casos gira em torno de aprimorar este aspecto. A técnica mais comum é por diferenças finitas, onde perturba-se cada entrada individualmente em torno do ponto de operação atual, e medem-se as saídas para se obter a estimativa dos gradientes. É sugerida a leitura de (MARCHETTI et al., 2016) para um maior detalhamento das principais técnicas disponíveis para a estimativa eficiente dos gradientes da planta.

Abordagens utilizando aproximação quadráticas também são propostas, onde destaca-se o trabalho inicial de (GAO; WENZEL; ENGELL, 2015b; GAO; WENZEL; ENGELL, 2015a; GAO; WENZEL; ENGELL, 2016), o qual é posteriormente explorado por (SINGHAL; FAULWASSER; BONVIN, 2015; SINGHAL et al., 2016). Elas são baseadas no uso de aproximações quadráticas locais sobre dados de iterações passadas para a função objetivo e restrições no ponto de operação. Dessa forma, os modificadores são estimados a partir destas funções e utilizados no Problema 2.4, ou a própria aproximação quadrática pode ser usada para o cálculo dos novos sinais de controle. Entretanto, elas sofrem dos mesmos problemas causados pela falta de robustez na estimativa dos gradientes da planta, além de necessitar do ajuste fino de muitos parâmetros. Para tentar superar estas limitação, técnicas que não dependem do cálculo de gradientes foram propostas na literatura. As principais serão descritas a seguir.

(NAVIA et al., 2015) propôs a reformulação do MA como um problema de otimização em dois níveis, sendo o primeiro responsável pela atualização dos modificadores e o segundo pelo cálculo dos sinais de controle ótimos. O ponto de operação ótimo é calculado pela solução do Problema 2.4, entretanto os modificadores são atualizados de forma bem diferente: em cada iteração do sistema RTO, somente uma iteração do problema de otimização para atualizar os modificadores é realizada, e posteriormente utilizada no problema estático. Se o algoritmo *Nelder-Mead* for utilizado, esse passo seria equivalente a uma atualização do *simplex*. Note que em nenhum momento há a necessidade da estimativa

dos gradientes, o que é uma grande vantagem desta técnica. O autor realizou uma análise bem detalhada, mostrando que a ela possui convergência similar às existentes, porém é mais robusta a ruídos nas medições. Os autores também validaram a técnica sobre um processo real (NAVIA et al., 2016), o que merece ser mencionado, visto que a maioria das abordagens MA da literatura são somente avaliadas com simulações.

(FERREIRA et al., 2018) propôs pela primeira vez uso de técnicas de aprendizado de máquina em sistemas RTO para eliminar a necessidade da estimativa dos gradientes. Neste trabalho, o autor propõe substituir os termos modificadores usados no Problema 2.4 pela modelagem da discrepância modelo-planta utilizando regressão por processos Gaussianos (GP) (WILLIAMS; RASMUSSEN, 2006), de forma que a função objetivo e restrições modificadas sejam localmente equivalentes à da planta. A cada iteração, a média, covariância e os hiper-parâmetros do GP são atualizados com novos dados medidos da função de custo e restrições, sem a necessidade de estimar os gradientes da planta. Através de simulações um reator simples, os autores mostram que a técnica proposta possui melhor desempenho do que a abordagem MA original (MARCHETTI; CHACHUAT; BONVIN, 2009) no que diz respeito à sua robustez aos ruídos de medição, o que é uma característica importante para aplicações reais.

Baseado no artigo anterior, (ANDERSSON; BRADFORD; IMSLAND, 2020) explora a aplicação de GPs para otimização em tempo-real de uma fazenda de energia eólica. Esta é uma aplicação muito interessante pois lida com um sistema distribuído de geração de energia, o que é bem diferente do encontrado na maioria da literatura. Em vez de modelar o sistema como um todo, o autor propôs identificar modelos de regressão GP para subsistemas da planta, que neste caso seriam as turbinas, os quais são posteriormente agregados para montar um modelo único que representa a discrepância entre a planta e o modelo. Através de estudos de casos de 3 e 10 turbinas em operação, o autor mostra que o aprendizado com regressão GP distribuída possui melhor desempenho em relação à modelagem única. Com os resultados desse artigo, é possível notar que o uso de GPs no contexto RTO é bem promissor, podendo ser aplicado em sistemas de energia e distribuídos.

Seguindo a mesma ideia introduzida por (FERREIRA et al., 2018), em (CHANONA et al., 2019) os autores apresentam duas propostas de algoritmos baseados em regiões de confiança com o objetivo de torná-lo ainda mais eficiente. O primeiro introduz uma região de confiança convencional em torno dos sinais de controle, cujo raio é ajustado de acordo com um modelo de regressão GP. O segundo define múltiplas regiões de confiança sobre a estimativa da variância do modelo de regressão GP, atuando diretamente sobre a função de custo e restrições. Utilizando o mesmo reator simulado do trabalho anterior, os autores mostram que os dois algoritmos apresentaram um melhor desempenho em relação à velocidade de convergência e capacidade de atenuar a violação de restrições durante as iterações. Vale a pena citar que a convergência global do método é posteriormente

comprovada por (SHUKLA et al., 2020).

No estado da arte, (RIO-CHANONA et al., 2020) investiga os benefícios de se integrar a abordagem descrita anteriormente com conceitos de otimização Bayesiana. Na técnica proposta, o autor incorpora o *framework* Bayesiano através da aplicação de funções de aquisição ao objetivo do Problema 2.4 modificado com GPs, mantendo o uso da técnica de regiões de confiança para lidar com as restrições. Um resultado importante foi a avaliação do potencial da abordagem para um problema mais complexo, mostrando que existem benefícios em se utilizar o modelo do processo em relação à uma abordagem puramente orientada aos dados. Esta técnica é bastante interessante, o levou ao estudo recente do uso de restrições probabilísticas como forma de torná-la mais eficiente no balanceamento entre exploração e refino durante o processo de otimização (PETSAGKOURAKIS; CHACHUAT; RIO-CHANONA, 2021).

Aplicações de sistemas RTO presentes na literatura são em sua maioria realizadas sobre problemas baseados em modelos simples de processos químicos (MARCHETTI et al., 2016), possuindo algumas poucas aplicações em sistemas de energia (SERRALUNGA; MUSSATI; AGUIRRE, 2013; COSTELLO; FRANÇOIS; BONVIN, 2015; ANDERSSON; BRADFORD; IMSLAND, 2020), rede de poços de petróleo (KRISHNAMOORTHY; FOSS; SKOGESTAD, 2016), compressores em paralelo (MILOSAVLJEVIC et al., 2020) e trajetória de robôs (MILOSAVLJEVIC et al., 2016), por exemplo. O problema mais comum é o reator de Williams-Otto (WILLIAMS; OTTO, 1960), o qual é usado em grande parte da literatura para comparar diferentes técnicas RTO. Ele é um processo contínuo operando em estado estacionário com três reações e dois sinais de controle, cujo objetivo é maximizar o lucro.

Embora menos usuais na literatura, processos descontínuos também estão presentes e podem ser resolvidos seguindo os mesmos princípios apresentados anteriormente (FRANCOIS; BONVIN, 2013). O problema mais comum é o reator de acetoacilação de pirrol (RUPPEN; BONVIN; RIPPIN, 1998). Ele é composto por três reações, considera um perfil de controle discretizado por três variáveis e seu objetivo é maximizar a produção de acetoacetilpirrol, respeitando os limites de composição final dos sub-produtos. Em geral este é um problema de otimização dinâmica, mas uma prática comum na literatura é discretizar as entradas em diferentes instantes de troca, transformando-o em um problema estático (BONVIN; SRINIVASAN; RUPPEN, 2001).

Uma deficiência encontrada na literatura foi a falta de problemas de *benchmark* com maior escala e de fácil acesso aplicáveis em problema RTO. O exemplo mais comum é o processo de *Tennessee Eastman* (DOWNS; VOGEL, 1993; BATHELT; RICKER; JELALI, 2015), o qual foi proposto para que a comunidade acadêmica pudesse estudar os desafios encontrados em ambientes industriais. Este modelo consegue representar a operação de uma planta inteira, possuindo cinco grandes operações unitárias: reator, condensador,

separador, compressor reciclador e removedor. Isso resulta em um total de 41 variáveis medidas e 12 controláveis, necessitando também do projeto de uma estrutura de controle para sua operação estável em malha fechada (RICKER, 1996). O trabalho de (GOLSHAN; PISHVAIE; BOOZARJOMEHRY, 2008) é um bom exemplo de aplicação do RTO TS aplicado sobre este problema.

Uma premissa de sistemas RTO é a existência de um algoritmo adequado para a solução do Problema 2.2. Na prática, *solvers* não lineares genéricos são geralmente utilizados (CÂMARA et al., 2017). Eles possuem implementações confiáveis e testadas de algoritmos determinísticos conhecidos, os quais são computacionalmente eficientes e capazes de lidar com problemas de larga-escala. Modelos de processo são em sua maioria na forma de equações diferenciais ou algébricas, exibindo com frequência comportamento não linear. Portanto, características indesejadas são introduzidas nos problemas de otimização, como descontinuidades e multimodalidade. Além disso, a escolha da solução inicial é importante para sua convergência: é conhecido que uma escolha inadequada torne impossível encontrar um bom ótimo local (KRUSE et al., 2016).

Neste cenário, *solvers* genéricos estão sujeitos a perder suas garantias de convergência global, podendo levar a efeitos inesperados no sistema RTO. Para a abordagem TS, (QUELHAS; JESUS; PINTO, 2013) mostrou como uma má escolha do ponto de partida e do método de otimização pode impactar no desempenho e convergência do sistema, considerando métodos determinísticos, estocásticos e híbridos. Entretanto, pouca ou nenhuma discussão está presente na literatura para este aspecto (QUELHAS; JESUS; PINTO, 2013). Apenas alguns outros trabalhos mencionam problemas numéricos relativos ao algoritmo de otimização (CUBILLOS; ACUÑA; LIMA, 2007; GOLSHAN; PISHVAIE; BOOZARJOMEHRY, 2008), sendo antigos e limitados ao método TS. Para aplicações do MA isso é praticamente inexistente, sendo difícil encontrar até qual algoritmo foi utilizado para a geração dos resultados.

Uma outra deficiência grave encontrada na literatura foi a falta de rigor metodológico para a comparação de sistemas RTO. Em geral, utiliza-se um modelo de *benchmark* simples com a inicialização do RTO de um ponto inicial arbitrário, realizando a comparação de somente uma execução até um número pré-determinado de iterações. Este tipo de experimento é muitas vezes executado considerando a presença de ruídos de medição, o que ignora os possíveis problemas numéricos descritos por (QUELHAS; JESUS; PINTO, 2013), comprometendo a reprodução e validade dos resultados apresentados. São muitas raras referências que realizam a execução de um experimento completo, destacando-se os artigos recentes de (ANDERSSON; IMSLAND, 2020; RIO-CHANONA et al., 2020), os quais deveriam ser utilizados como referência para trabalhos futuros.

## 3 Fundamentação Teórica

### 3.1 Introdução

Este capítulo tem como objetivo apresentar os principais fundamentos teóricos envolvidos na otimização em tempo-real. Por se tratar de um problema multi-disciplinar, é dado um enfoque maior na teoria de otimização não linear e os principais algoritmos utilizados neste trabalho. Além disso, é realizada uma introdução aos Processos Gaussianos e à otimização Bayesiana.

### 3.2 Otimização não linear

Um problema de programação não linear é geralmente definido como (BAZARAA; SHERALI; SHETTY, 2013):

$$\begin{aligned}
 & \underset{\mathbf{x}}{\text{minimizar}} && f(\mathbf{x}) \\
 & \text{sujeito a} && g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\
 & && h_j(\mathbf{x}) = 0, \quad j = 1, \dots, l \\
 & && \mathbf{x} \in \mathcal{X}
 \end{aligned} \tag{3.1}$$

onde  $\mathbf{x}$  é um vetor com  $n$  componentes, também conhecidas por variáveis de decisão;  $f(\mathbf{x})$  é a função objetivo,  $\mathbf{g}(\mathbf{x})$  e  $\mathbf{h}(\mathbf{x})$  são os vetores das funções de restrição de igualdade e desigualdade, respectivamente.  $\mathcal{X}$  é um sub-conjunto de  $\mathbb{R}^n$  que define o espaço de busca das variáveis de decisão. Quando as restrições de desigualdade e igualdade não são definidas, o Problema 3.1 se torna irrestrito. Nosso problema consiste em encontrar um vetor  $\mathbf{x}^*$  que satisfaça todas as restrições e possua objetivo inferior às demais soluções viáveis dentro do espaço de busca  $\mathcal{X}$ .

Suponha que  $f$ ,  $\mathbf{g}$  e  $\mathbf{h}$  sejam diferenciáveis e que para uma solução viável qualquer  $\mathbf{x}$ ,  $\nabla g_i(\mathbf{x}) \in I = \{i : g_i(\mathbf{x}) = 0\}$  e  $\nabla h_j(\mathbf{x})$ ,  $j = 1, \dots, l$  são linearmente independentes. Se  $\mathbf{x}$  é um ótimo local, então existem vetores  $\boldsymbol{\mu}$  e  $\boldsymbol{\lambda}$  únicos tal que (BAZARAA; SHERALI; SHETTY, 2013):

$$\nabla f(\mathbf{x}) - \sum_{i=1}^m \mu_i \nabla g_i(\mathbf{x}) + \sum_{j=1}^l \lambda_j \nabla h_j(\mathbf{x}) = \mathbf{0} \tag{3.2}$$

$$\mu_i g_i(\mathbf{x}) = 0, \quad i = 1, \dots, m \tag{3.3}$$

$$\mu_i \geq 0, \quad i = 1, \dots, m \tag{3.4}$$

Estas são as condições necessárias de otimalidade de primeira-ordem (NCO), também conhecidas como as condições de Karush-Kuhn-Tucker (KKT). Se  $f(\mathbf{x})$  é uma função pseudo-convexa,  $\mathbf{g}(\mathbf{x})$  quasi-convexas e  $\mathbf{h}(\mathbf{x})$  funções lineares, então elas se tornam também condições suficientes (SCO), significando que  $\mathbf{x}$  também é um ótimo global (BAZARAA; SHERALI; SHETTY, 2013). Em geral, estas premissas não são sempre válidas, de forma que informações adicionais são necessárias para avaliar a otimalidade local de  $\mathbf{x}$ . Isso levou ao desenvolvimento das condições de segunda-ordem, as quais estabelecem que se a Hessiana da Lagrangeana associada é positiva definida para todo  $\mathbf{x} \in \mathcal{X}$  e as condições de primeira ordem são satisfeitas, logo  $\mathbf{x}$  é um ótimo global (BAZARAA; SHERALI; SHETTY, 2013). Caso seja positiva definida em apenas uma vizinhança de  $\mathbf{x}$ , será um ótimo local nessa vizinhança (BAZARAA; SHERALI; SHETTY, 2013).

Vários métodos estão disponíveis na literatura para resolver o Problema 3.1. Abordagens determinísticas geralmente baseiam-se em informações dos gradientes ou matriz Hessiana das funções de objetivo e restrições para gerar soluções candidatas em cada iteração. Também existem heurísticas de busca direta que não precisam de informações dos gradientes, que são uma alternativa quando estes não estão disponíveis ou são difíceis de avaliar. As técnicas determinísticas são geralmente rápidas e podem lidar com problemas de grande escala. No entanto, problemas reais geralmente apresentam comportamento não-convexo, de modo que é muito provável que não consigam encontrar o ótimo global. Além disso, elas exigem uma estimativa inicial para as variáveis de decisão, para o qual uma escolha ruim pode tornar impossível até mesmo encontrar uma boa solução local (KRUSE et al., 2016).

Para lidar com essa limitação, os métodos de otimização estocásticos são uma boa alternativa se os recursos computacionais não forem uma restrição, especialmente os algoritmos evolutivos. Inspiradas em conceitos biológicos, essas técnicas evoluem a partir de uma população inicial baseada em operadores de variação e seleção. Em cada iteração, ou geração, operadores de recombinação e mutação são aplicados para criar diversidade na população, facilitando a exploração do espaço de busca. Em seguida, são selecionados os indivíduos mais aptos, melhorando a qualidade média das soluções na população. Esse processo geralmente leva a uma melhor adequação em cada geração e, se houver tempo computacional suficiente disponível, a convergência para o ótimo global pode ser alcançada. Muitas estratégias evolutivas estão disponíveis, como os bastante conhecidos Algoritmos Genéticos (GA) (DAVIS, 1991).

Como a escolha do método depende do problema, uma descrição completa dos algoritmos disponíveis está além do escopo deste trabalho. Para um detalhamento da teoria e dos algoritmos determinísticos disponíveis, é sugerida a leitura de (BAZARAA; SHERALI; SHETTY, 2013; RAO, 2019), por exemplo. Já para um estudo mais detalhado das heurísticas evolutivas disponíveis, (EIBEN; SMITH, 2015) é um bom ponto de partida



para o leitor interessado. Nas seções a seguir são brevemente apresentados os algoritmos de otimização utilizados para o desenvolvimento do trabalho.

### 3.2.1 O algoritmo Sequential Quadratic Programming

De forma geral, dado um vetor  $\mathbf{x}_0$  de inicialização, a solução do Problema 3.1 é obtida iterativamente pela seguinte relação:

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad (3.5)$$

onde  $\alpha_k \in (0, \infty]$  é o tamanho do passo e  $\mathbf{d}_k \in \mathbb{R}^n$  é uma direção de busca. O tamanho do passo pode ser calculado usando técnicas de otimização unidimensional, como o algoritmo da seção áurea ou biseção. A forma como  $\mathbf{d}_k$  é calculado em cada iteração define as características de convergência e premissas dos diferentes algoritmos propostos na literatura (BAZARAA; SHERALI; SHETTY, 2013).

No *Sequential Quadratic Programming* (SQP), a direção de busca é calculada através da solução de um sub-problema de otimização não linear, formulado como uma aproximação quadrática da Lagrangiana associada e uma aproximação linear das restrições  $\mathbf{g}$  e  $\mathbf{h}$  (KRAFT et al., 1988):

$$\begin{aligned} \underset{\mathbf{d}}{\text{minimizar}} \quad & \frac{1}{2} \mathbf{d}^T H_k \mathbf{d} + \nabla f(\mathbf{x}_k) \mathbf{d} \\ \text{sujeito a} \quad & \nabla g_i(\mathbf{x}_k)^T \mathbf{d} + g_i(\mathbf{x}_k) \leq 0, \quad i = 1, \dots, m \\ & \nabla h_j(\mathbf{x}_k)^T \mathbf{d} + h_j(\mathbf{x}_k) = 0, \quad j = 1, \dots, l \end{aligned} \quad (3.6)$$

onde  $H$  é uma aproximação positiva definida da matriz Hessiana, a qual pode ser calculada com qualquer abordagem quasi-Newton, como a BFGS (KRAFT et al., 1988).

### 3.2.2 O algoritmo Differential Evolution

O algoritmo *Differential Evolution* (DE) é uma meta-heurística baseada em populações bastante utilizada na literatura, principalmente devido à sua simplicidade e seu excelente desempenho quando aplicado em problemas de diversos domínios (PANT et al., 2020). Sua principal diferença em relação às demais heurísticas está no seu operador de mutação diferencial. Dado um indivíduo  $i$  de uma população com  $P$  soluções candidatas, um vetor mutante  $\mathbf{v}_i$  é criado adicionando uma perturbação proporcional à diferença de dois outros membros aleatórios desta população:

$$\mathbf{v}_i = \mathbf{x}_a + F(\mathbf{x}_b - \mathbf{x}_c), \forall i \in [1, \dots, P] \quad (3.7)$$

onde o fator de escala  $F > 0$  é um número real que controla a taxa com que a população evolui,  $\mathbf{x}_a$ ,  $\mathbf{x}_b$  e  $\mathbf{x}_c$  são indivíduos selecionados aleatoriamente da população em sua versão clássica (EIBEN; SMITH, 2015).

Em seguida os indivíduos da população original são recombinados com os da mutante, produzindo uma população de  $O$  descendentes. Na sua versão clássica, emprega-se a recombinação discreta com probabilidade  $C_r \in [0, 1]$ . Uma particularidade introduzida neste operador consiste que para um índice aleatório, o alelo de um descendente sempre será do indivíduo mutante. Isso garante com que pelo menos um dos alelos de uma solução descendente seja herdado de uma mutação (EIBEN; SMITH, 2015).

O último passo consiste na seleção dos indivíduos sobreviventes. Para cada descendente  $\mathbf{o}_i$ , seu valor de função objetivo  $f(\mathbf{o}_i)$  é calculado e comparado com o valor da respectiva solução corrente  $f(\mathbf{x}_i)$ . Na versão clássica, se  $f(\mathbf{o}_i) \leq f(\mathbf{x}_i)$ , então o indivíduo  $\mathbf{o}_i$  substitui o  $\mathbf{x}_i$  na população. Caso contrário,  $\mathbf{o}_i$  é descartada e  $\mathbf{x}_i$  sobrevive. Este processo se repete até que um critério de parada definido pelo usuário seja atingido.

Diversas variações deste algoritmo estão presentes na literatura (PANT et al., 2020) e em geral são representadas pela notação *base/d/rec*, onde *base* se refere a como o vetor de base  $\mathbf{x}_a$  é escolhido,  $d$  o número de vetores de diferenças ( $\mathbf{x}_b - \mathbf{x}_c$ ) e *rec* identifica o operador de recombinação utilizado. Sua versão clássica é representada por *rand/1/bin*, por exemplo.

### 3.2.3 O algoritmo Nelder-Mead Simplex

Inicialmente proposto em 1965 (NELDER; MEAD, 1965), o algoritmo *Nelder-Mead Simplex* (NM) pertence à classe de técnicas de busca direta, ou seja, ele é capaz de lidar com problemas multidimensionais irrestritos sem a necessidade de cálculo de derivadas. Essa característica o torna adequado para otimizar funções não-convexas, sendo bastante utilizado em problemas de estimativa de parâmetros e controle de processos (SINGER; NELDER, 2009). Entretanto, vale ressaltar que sua convergência é comprovada somente para problemas convexos (LAGARIAS et al., 1998).

O primeiro passo consiste na construção de um simplex inicial, usualmente pela geração de  $n + 1$  vértices  $\mathbf{x}_1, \dots, \mathbf{x}_{n+1}$  em torno de um ponto inicial. Feito isso, uma iteração deste algoritmo consiste nos seguintes passos (SINGER; NELDER, 2009):

1. **Ordenação:** Determina os índices do pior  $w$ , segundo pior  $s$  e melhor  $l$  vértice no simplex atual de acordo com os valores da função objetivo. Para a minimização, isso equivale a:

$$f(\mathbf{x}_w) > f(\mathbf{x}_s) > \dots > f(\mathbf{x}_l)$$

2. **Centroide:** Calcula o centroide da melhor face do simplex, ou seja, aquela oposta ao pior vértice:

$$\mathbf{c} = \frac{1}{n} \sum_{i \neq w} \mathbf{x}_{i,j}$$

3. **Transformação:** Computa o novo simplex, inicialmente tentando substituir o pior vértice por um ponto melhor usando as operações de reflexão, expansão ou contração em relação à melhor face. Se o novo ponto resultante melhorar a função para pelo menos uma destas operações, ele então substitui o pior ponto. Caso contrário, o simplex é encolhido em direção ao melhor ponto.

A definição completa das operações de transformação mencionadas anteriormente serão omitidas por brevidade, mas podem ser encontradas em (SINGER; NELDER, 2009) por exemplo. Através destes passos simples, o NM é geralmente capaz de rapidamente produzir bons resultados para problemas complexos, sendo bastante popular na literatura.

### 3.3 Processos Gaussianos

Quando estamos interessados em aprender uma função desconhecida com base nos dados disponíveis, uma abordagem comum são os métodos paramétricos. Eles são baseados em uma função definida por um conjunto fixo de parâmetros, os quais são ajustados para descrever da melhor forma os dados durante uma fase de aprendizagem. Embora sejam fáceis de entender e interpretar, modelos paramétricos simples podem não ter poder preditivo para dados complexos (WILLIAMS; RASMUSSEN, 2006). Nesta situação, o uso de modelos não-paramétricos baseados em métodos de *kernel*, como os Processos Gaussianos (GPs), são uma alternativa flexível e poderosa (WILLIAMS; RASMUSSEN, 2006).

O objetivo dos GPs é tentar descrever uma função desconhecida  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  para um conjunto de de observações ruidosas  $y = f(\mathbf{x}) + \nu$ , onde  $\nu \sim \mathcal{N}(0, \sigma_\nu^2)$  é um ruído de medição que segue uma distribuição Gaussiana com média zero e variância  $\sigma_\nu^2$  (geralmente desconhecida). Eles podem ser entendidos como uma generalização da distribuição Gaussiana multivariada, significando que suas predições possuem não só uma estimativa da média, mas também de sua variância associada (WILLIAMS; RASMUSSEN, 2006). Em um ponto arbitrário  $\mathbf{x}$ , podemos definir a função  $f$  como:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^T)) \quad (3.8)$$

onde  $m(\cdot)$  é função de média, a qual reflete o valor esperado de  $\mathbf{x}$ ;  $k(\cdot, \cdot)$  é a função de covariâncias (ou *kernel*), a qual modela a dependência entre as entradas em diferentes pontos. A predição é realizada utilizando o modelo GP como informação *a priori* para inferência Bayesiana. Uma descrição detalhada deste processo será omitida por brevidade, mas pode ser encontrada em (WILLIAMS; RASMUSSEN, 2006), por exemplo.

Na prática, as funções exatas de média e covariância são geralmente desconhecidas, sendo portanto parametrizadas por um conjunto de hiper-parâmetros. Uma escolha comum

é assumir a média como um valor constante e utilizar um *kernel* Gaussiano (RBF):

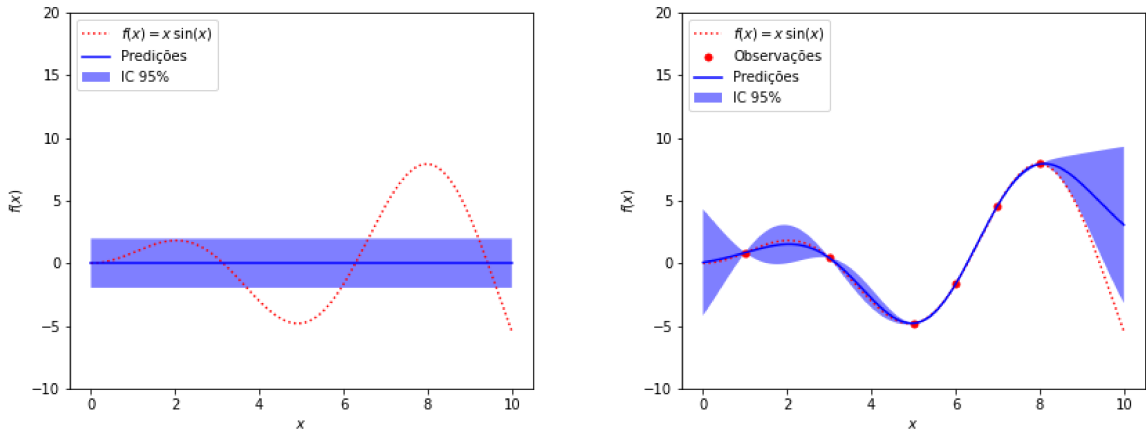
$$\begin{aligned} m(\mathbf{x}) &:= c \\ k(\mathbf{x}, \mathbf{x}') &:= \sigma_n^2 \exp\left(\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Lambda (\mathbf{x} - \mathbf{x}')\right) \end{aligned} \quad (3.9)$$

onde  $\sigma_n^2$  é a magnitude da covariância e  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  é a matriz de escala. Dado um conjunto de  $N$  amostras  $\mathbf{y} = [y_1, \dots, y_N]^T$  coletados para uma matriz de entrada  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ , a distribuição prevista de  $f$  em um ponto arbitrário  $\mathbf{x}$  para este *kernel* será Gaussiana (WILLIAMS; RASMUSSEN, 2006):

$$f(\mathbf{x})|\mathbf{X}, \mathbf{y} \sim \mathcal{N}(\mu_f(\mathbf{x}), \sigma_f^2(\mathbf{x})) \quad (3.10)$$

onde  $\mu_f$  and  $\sigma_f^2$  são as funções de média e covariância posteriores, respectivamente. Um exemplo de sua aplicação pode ser visto na Figura 4.

O ajuste dos hiper-parâmetros aos dados pode ser feito com a maximização do logaritmo da função de verossimilhança marginal (MYUNG, 2003). Embora esta técnica inclua uma penalização que previne o sobre-ajuste em sua formulação (WILLIAMS; RASMUSSEN, 2006), este fenômeno ainda pode ocorrer nesta etapa devido à escolha de um *kernel* inadequado, ou se o conjunto de dados é pequeno e existam amostras muito concentradas em uma região do espaço (HAWKINS, 2004; MOHAMMED; CAWLEY, 2017).



(a) Predição com informação a *priori* de média  $c = 0$  e *kernel* RBF com  $\Lambda = [1]$  e  $\sigma_n^2 = 1$  (b) Predição com informação das amostras e ajuste dos parâmetros do *kernel*

Figura 4 – Exemplo de predições com processos Gaussianos para a função  $f(x) = x \sin(x)$ . As áreas azuis representam o intervalo de confiança de 95% para a média.

### 3.4 Otimização Bayesiana

A Otimização Bayesiana (OB) é uma técnica utilizada em problemas cujo objetivo é encontrar o ótimo global de uma função  $f(\mathbf{x})$  dentro de um domínio  $\mathcal{X}$ , a qual apresenta as seguintes propriedades (GELBART, 2015):

1. **Caixa-preta:**  $f$  só pode ser avaliada de forma pontual, sem acesso às suas derivadas.
2. **Avaliação cara:** amostrar  $f$  para qualquer  $\mathbf{x}$  dentro do seu domínio resulta em um tempo computacional elevado ou um custo financeiro associado.
3. **Avaliação ruidosa:** pode ser somente possível observar o seu valor real corrompido por algum tipo de ruído.

É importante ressaltar que problemas apresentando estas características são muito difíceis de serem resolvidos com OB quando a sua dimensionalidade é muito alta. De acordo com (FRAZIER, 2018), aplicações de sucesso possuem até 20 variáveis de decisão. Como seu objetivo é encontrar um ótimo global, existe então a necessidade de se balancear automaticamente uma busca global (explorar todo o domínio) e local (refino de soluções potenciais) durante suas iterações (GELBART, 2015). A OB pode ser decomposta em dois componentes principais: um modelo estatístico Bayesiano  $\mathcal{M}$  para representar  $f$  e uma função de aquisição  $\alpha$  para decidir qual será a próxima solução avaliada. Dado um conjunto de dados iniciais  $\mathcal{D}$  composto, por exemplo, de pontos amostrados uniformemente em  $\mathcal{X}$ , o procedimento completo é descrito no [Algoritmo 1](#).

---

**Algoritmo 1:** Otimização Bayesiana irrestrita

---

**Entrada:** objetivo  $f$ , função de aquisição  $\alpha$ , domínio  $\mathcal{X}$ , modelo  $\mathcal{M}$  treinado sobre os dados iniciais  $\mathcal{D}$

1 **repita**

2     Maximizar a função de aquisição:  $\hat{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}, \mathcal{M})$  ;

3     Avaliar a função objetivo:  $\hat{y} = f(\hat{\mathbf{x}})$  ;

4     Incluir os novos dados:  $\mathcal{D} = \mathcal{D} \cup \{(\hat{\mathbf{x}}, \hat{y})\}$  ;

5     Ajustar o modelo  $\mathcal{M}$  sobre os dados  $\mathcal{D}$  atualizados

6 **até atingir critério de convergência;**

**Saída:** Uma solução ótima  $\mathbf{x}^*$

---

GPs são geralmente utilizados para modelar a função objetivo, sendo seu ajuste feito conforme descrito na Seção 3.3. A função de aquisição determina qual o ganho esperado ao se avaliar a função objetivo em  $\mathbf{x}$ , baseado na distribuição posterior de  $f$  estimada pelo modelo atual. Diferentes funções de aquisição existem na literatura, sendo a *Expected Improvement* (EI) uma das mais comuns (FRAZIER, 2018). Proposta inicialmente por (JONES; SCHONLAU; WELCH, 1998), ela mede a quantidade de melhora esperada pela observação de  $f$  em relação à uma referência  $\eta$ :

$$\alpha_{EI}(\mathbf{x}) = \mathbb{E}_y[\max(0, \eta - y)] = \int_{-\infty}^{\infty} \max(0, \eta - y)p(y|\mathbf{x})dy \quad (3.11)$$

onde  $p(y|\mathbf{x})$  é a distribuição posterior preditiva de  $f$  em  $\mathbf{x}$ . A referência  $\eta$  é geralmente escolhida como a melhor solução encontrada até o momento. Quando a distribuição

posterior sobre o modelo é Gaussiana (que é o caso dos GPs), a EI possui a seguinte expressão fechada (GELBART, 2015):

$$\alpha_{EI}(\mathbf{x}) = \sigma(\mathbf{x})(z(\mathbf{x})\Phi(z(\mathbf{x})) + \varphi(z(\mathbf{x}))) \quad (3.12)$$

onde  $z(\mathbf{x}) := \frac{\eta - \mu(\mathbf{x})}{\sigma(\mathbf{x})}$ ;  $\mu(\mathbf{x})$  e  $\sigma(\mathbf{x})$  são a média e desvio padrão da distribuição posterior em  $\mathbf{x}$ , respectivamente;  $\Phi(\cdot)$  é a função de distribuição cumulativa (CDF) e  $\phi(\cdot)$  a função densidade de probabilidade (PDF), ambas considerando a distribuição normal padrão. Um exemplo de sua aplicação pode ser visto na Figura 5.

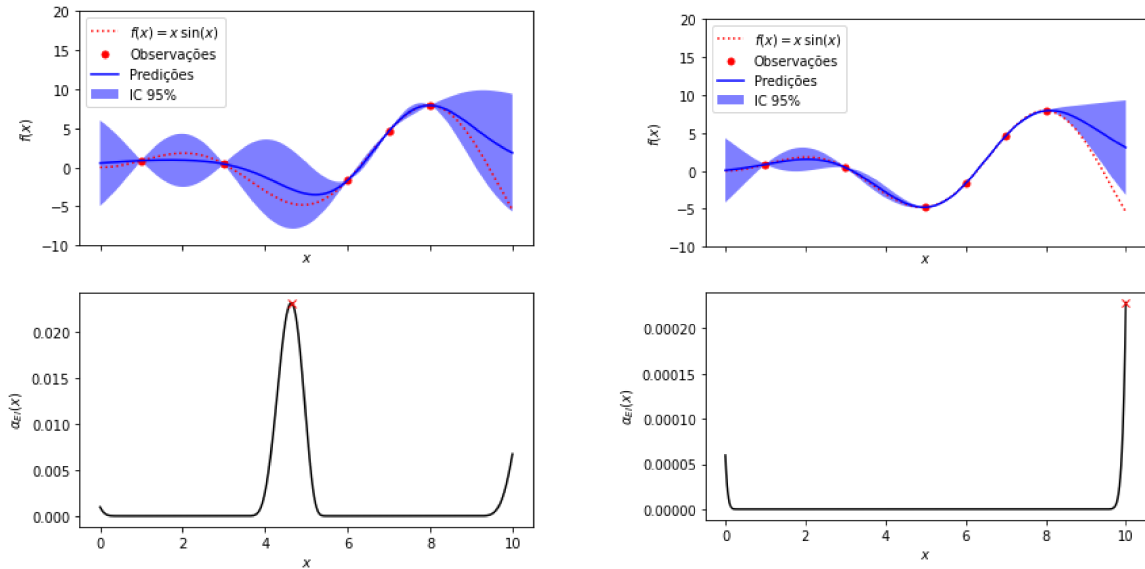


Figura 5 – Exemplo de otimização Bayesiana com a função de aquisição EI para estimar o mínimo da função  $f(x) = x \sin(x)$ . Numa primeira iteração (a), note o máximo do EI corresponde a um ponto bem próximo do mínimo da função, de acordo com o modelo GP das observações. Este ponto é posteriormente amostrado (b), onde podemos ver que a função EI sugere explorar uma região ainda não avaliada.

A discussão anterior assume que a avaliação de  $f$  não possui ruídos de medição. Tal premissa é geralmente violada na prática, a qual é tratada pelo ajuste da variância como um hiper-parâmetro dos modelos GP (FRAZIER, 2018). Além disso, o EI e outras funções de aquisição da literatura são por definição limitadas a problema de otimização irrestritos. Quando as restrições retornam valores contínuos quando avaliadas, podemos utilizar uma modelagem via GPs similar à realizado para a função objetivo. Para uma discussão sobre restrições mais complexas, o leitor pode se referir a (GELBART, 2015), por exemplo.

Uma forma direta de se adicionar restrições à OB consiste em uma pequena modificação na função de aquisição EI. Primeiro, ampliamos a definição de  $\eta$  para ser o ponto viável com menor valor de objetivo observado. Depois, associamos uma melhoria nula à todos os pontos inviáveis (GARDNER et al., 2014). Ou seja, estamos definindo a

melhora somente quando as restrições são respeitadas. Assumindo independência entre elas, define-se então a técnica conhecida por *Expected Improvement with Constraints* (EIC) (GELBART, 2015):

$$\alpha_{EIC}(\mathbf{x}) = \alpha_{EI}(\mathbf{x}) \prod_{k=1}^{n_g} P(g_k(\mathbf{x}) \leq 0) \quad (3.13)$$

onde  $P(\cdot)$  é a probabilidade da restrição ser respeitada. Quando modelamos as restrições via GPs sua probabilidade segue o formato de uma CDF Gaussiana, resultado na seguinte equação fechada similar ao EI:

$$\alpha_{EIC}(\mathbf{x}) = \sigma_f(\mathbf{x})(z_f(\mathbf{x})\Phi(z_f(\mathbf{x})) + \varphi(z_f(\mathbf{x}))) \prod_{k=1}^{n_g} \Phi\left(\frac{\mu_k(\mathbf{x})}{\sigma_k(\mathbf{x})}\right) \quad (3.14)$$

onde  $\mu_k$  e  $\sigma_k^2$  são a média e variância previstas pelo modelo GP para a  $k$ -ésima restrição;  $\mu_f$  e  $\sigma_f^2$  representam a função objetivo, sendo  $z_f(\mathbf{x}) = z(\mathbf{x})$ . Uma discussão de técnicas mais avançadas para problemas OB restrita pode ser encontrada em (GELBART, 2015).

## 4 Metodologia

### 4.1 Introdução

Este capítulo tem como objetivo apresentar a metodologia utilizada durante a realização deste trabalho. Inicialmente são descritos o *Modifier Adaptation* com Processos Gaussianos e a nova abordagem proposta via otimização Bayesiana restrita com função de aquisição EIC. Em seguida, é feita a descrição dos experimentos desenhados para se comparar o desempenho de sistemas RTO. Para auxiliar na replicação dos resultados, as diversas etapas da implementação são descritas, expondo as dificuldades esperadas. Finalmente, define-se os modelos de *benchmark* que serão utilizados nos estudos de caso.

### 4.2 Modifier Adaptation com Processos Gaussianos

Conforme discutido no [Capítulo 2](#), o maior desafio do MA é estimar os gradientes da planta para o cálculo dos modificadores. A partir da discussão anterior, é possível notar que, mantendo o respeito aos critérios de otimalidade, podemos utilizar qualquer representação para a discrepância entre a planta e o modelo. Portanto, o uso de técnicas de regressão não linear como representação alternativa aos modificadores é uma abordagem válida, a qual foi proposta e validada pela primeira vez por ([FERREIRA et al., 2018](#)).

Neste método, o qual será referido neste trabalho como MA-GP, o autor utiliza regressão por GPs para modelar a discrepância planta-modelo das funções objetivo e de restrição utilizando os dados medidos em estado estacionário. A justificativa da escolha desta técnica de regressão se deve às suas características descritas na [Seção 3.3](#) e o fato de já ser usada com sucesso na literatura em diversas aplicações de controle e otimização de processos ([KOCIJAN, 2016](#)). O conceito principal do MA-GP é substituir os modificadores do Problema [2.4](#) por GPs:

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimizar}} \quad & G_0(\mathbf{u}) + \mathcal{GP}_k^{(G_0^p - G_0)}(\mathbf{u}, \mathbf{U}_k, \mathbf{y}_k) \\ \text{sujeito a} \quad & G_i(\mathbf{u}) + \mathcal{GP}_k^{(G_i^p - G_i)}(\mathbf{u}, \mathbf{U}_k, \mathbf{y}_k) \leq 0, \quad i = 1, \dots, n_g \\ & \mathbf{u} \in \mathcal{U} \end{aligned} \tag{4.1}$$

onde a discrepância do objetivo e restrições são modeladas por  $\mathcal{GP}^{(G_i^p - G_i)} \in \mathbb{R}, i = 0, \dots, n_g$ ;  $(\mathbf{U}_k, \mathbf{y}_k)$  são os dados de entrada-saída para treinamento dos GPs na  $k$ -ésima iteração. Para evitar correções excessivas quando estiver distante do ótimo da planta e reduzir a sensibilidade à ruídos de medição ([MARCHETTI; CHACHUAT; BONVIN, 2009](#)), é recomendado aplicar um filtro exponencial de primeira-ordem (passa-baixas) ([JR, 1985](#)) à solução ótima



$\mathbf{u}_{k+1}^*$  encontrada para o Problema 4.1 antes de sua aplicação na planta (FERREIRA et al., 2018). Isso pode ser realizado a partir da seguinte equação (MARCHETTI et al., 2016):

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{K}(\mathbf{u}_{k+1}^* - \mathbf{u}_k) \quad (4.2)$$

onde  $\mathbf{u}_{k+1}$  é a nova solução a ser avaliada e  $\mathbf{K} = \text{diag}(k_1, \dots, k_{n_u})$ ,  $k_i \in (0, 1]$  é uma matriz de ganho.

Com o uso dos GPs, existe agora a necessidade de escolher o conjunto de dados de treinamento, especialmente para a inicialização do método. Existe bastante liberdade para esta escolha inicial, sendo uma prática comum na literatura o uso do MA original (FERREIRA et al., 2018; CHANONA et al., 2019). Após a inicialização, a seleção dos dados deve ser feita de forma a prevenir problemas numéricos e o sobre-ajuste dos GPs aos dados. É sugerido portanto limitar o tamanho do conjunto de dados de treinamento, considerando, por exemplo, as últimas  $n_K$  iterações ou os pontos de operação mais próximos de  $\mathbf{u}_{k+1}$  (FERREIRA et al., 2018). Além disso, uma outra boa prática consiste em não incluir a solução  $\mathbf{u}_{k+1}$  se ela está dentro de um determinado raio em relação às amostras existentes em  $\mathbf{U}_k$  (CHANONA et al., 2019). A rotina completa é apresentada no Algoritmo 2.

---

**Algoritmo 2:** *Modifier Adaptation* com Processos Gaussianos

---

**Entrada:** GPs treinados sobre os dados iniciais ( $\mathbf{U}_0, \mathbf{y}_0$ )

1 **repita**

2     Encontrar a solução do problema 4.1 ;

3     Filtrar a solução encontrada e obter  $\mathbf{u}_{k+1}$  (Equação 4.2) ;

4     Obter as medições das função objetivo e restrições  $G_i^p(\mathbf{u}_{k+1})$ ,  $i = 0, \dots, n_g$  e estimar a diferença em relação ao modelo  $\mathbf{y}_{k+1} := (G_{k+1}^p - G_{k+1})$ ;

5     Atualizar os dados de treinamento ( $\mathbf{U}_{k+1}, \mathbf{y}_{k+1}$ ) com as medidas em  $\mathbf{u}_{k+1}$  ;

6     Ajustar os GPs sobre o conjunto de dados atualizado

7 **até atingir critério de convergência;**

---

### 4.3 Modifier Adaptation com função de aquisição EIC

Ao observar os Algoritmos 1 e 2, nota-se que ambos possuem como objetivo otimizar uma função desconhecida a partir de um aproximação inicial, cuja incerteza é iterativamente reduzida de acordo com uma heurística de otimização. As seguintes características comuns entre o RTO e a OB são observadas:

- **Ambos visam otimizar uma função desconhecida e com alto custo de avaliação:** embora seja genérica para a OB, no caso do RTO nossa função é geralmente um processo industrial complexo, do qual temos conhecimento limitado do seu comportamento e altos custos econômicos envolvidos;

- **Ambos realizam a adaptação de um modelo que aproxima essa função:** na OB escolhe-se uma família de funções que vai sendo ajustada numa abordagem puramente orientada a dados. No RTO, existe um modelo determinístico aproximado desta função, cujos parâmetros podem ser ajustados ou modelada a discrepância entre eles a partir dos dados medidos;
- **Ambos calculam a próxima solução a ser avaliada através de um problema de otimização:** no RTO podemos otimizar diretamente o modelo, uma vez que ele fornece uma estimativa da função desconhecida em todo o seu domínio. Entretanto, na OB é necessária a otimização de funções de aquisição, pois a informação do modelo será limitada aos dados em que foi treinado.

Portanto, o RTO pode ser abstraído como uma aplicação de OB, para a qual existe um modelo determinístico aproximado da função desconhecida. Isso fica ainda mais evidente para o MA-GP, uma vez que ele modela a discrepância modelo-planta com GPs. A integração entre o MA-GP e OB foi inicialmente investigada por (RIO-CHANONA et al., 2020), reforçando a conclusão anterior. Na técnica proposta, o autor incorpora o *framework* Bayesiano através da aplicação de uma função de aquisição ao objetivo do Problema 4.1. Para lidar com as restrições da planta, ele propôs o uso da técnica de regiões de confiança (SHUKLA et al., 2020). Um resultado importante de (RIO-CHANONA et al., 2020) foi mostrar que existem benefícios em se utilizar o modelo aproximado do processo em relação à uma abordagem puramente Bayesiana.

De posse destes resultados, surge a hipótese de utilizar o *framework* Bayesiano restrito ao invés de regiões de confiança para lidar com as restrições. Para avaliá-la, é proposto o uso da função de aquisição EIC descrita no Capítulo 3. Embora existam funções mais eficientes (GELBART, 2015), o seu uso é justificado pela facilidade de implementação e entendimento. Para este fim, é necessário ajustar a Equação 3.14 para incorporar os GPs e modelo do processo:

$$\alpha_{EIC}(\mathbf{x}, \mathcal{GP}) = \sigma_f^{\mathcal{GP}}(\mathbf{x})(z_f^{\mathcal{GP}}(\mathbf{x})\Phi(z_f^{\mathcal{GP}}(\mathbf{x})) + \varphi(z_f^{\mathcal{GP}}(\mathbf{x}))) \prod_{k=1}^{n_g} \Phi\left(\frac{\mu_{G,k}^{\mathcal{GP}}(\mathbf{x})}{\sigma_{G,k}^{\mathcal{GP}}(\mathbf{x})}\right) \quad (4.3)$$

onde  $\sigma_f^{\mathcal{GP}}$  é o desvio padrão estimado pelo modelo GP da discrepância da função objetivo; seguindo o mesmo princípio,  $\sigma_{G,k}^{\mathcal{GP}}$  é o desvio padrão do modelo GP para a  $k$ -ésima restrição. O termo  $z_f^{\mathcal{GP}}$  é definido como:

$$z_f^{\mathcal{GP}}(\mathbf{x}) = \frac{\eta - (G_0(\mathbf{x}) + \mu^{\mathcal{GP}(f_p-f)}(\mathbf{x}))}{\sigma^{\mathcal{GP}(f_p-f)}(\mathbf{x})} \quad (4.4)$$

e  $\mu_k^{\mathcal{GP}}(\mathbf{x})$ ,  $k = 1, \dots, n_g$ , é dado por:

$$\mu_k^{\mathcal{GP}}(\mathbf{x}) = G_k(\mathbf{x}) + \mu^{\mathcal{GP}(G_p, k - G_k)}(\mathbf{x}) \quad (4.5)$$

Incorporando estas modificações no Problema 4.1, o MA-GP com função de aquisição EIC (MA-GP-EIC) pode ser definido pelo Problema 4.6. Note que ele apresenta a grande vantagem de ser irrestrito, simplificando a solução do problema de otimização do método proposto por (RIO-CHANONA et al., 2020). Além disso, ela remove o ajuste dos hiper-parâmetros necessários para construção das regiões de confiança. No Algoritmo 3 é apresentada a rotina completa para execução do sistema RTO com esta abordagem. As técnicas de filtragem das soluções do problema de otimização (Equação 4.2), e de seleção e descarte de amostras para o treinamento dos GPs serão mantidas idênticas ao MA-GP. Além disso, a referência  $\eta$  para a função de aquisição EIC será a melhor solução factível encontrada até o momento.

$$\begin{aligned} & \underset{\mathbf{u}}{\text{maximizar}} && \alpha_{EIC}(\mathbf{u}, \mathcal{GP}) \\ & \text{sujeito a} && \mathbf{u} \in \mathcal{U} \end{aligned} \quad (4.6)$$

---

**Algoritmo 3:** *Modifer Adaptation* com função de aquisição EIC

---

**Entrada:** GPs treinados sobre os dados iniciais  $(\mathbf{U}_0, \mathbf{y}_0)$ , referência inicial  $\eta$

- 1 **repita**
- 2   | Encontrar a solução do problema 4.6 ;
- 3   | Filtrar a solução encontrada e obter  $\mathbf{u}_{k+1}$  (Equação 4.2) ;
- 4   | Obter as medições das função objetivo e restrições  $G_i^p(\mathbf{u}_{k+1})$ ,  $i = 0, \dots, n_g$  e  
     | estimar a diferença em relação ao modelo  $\mathbf{y}_{k+1} := (G_{k+1}^p - G_{k+1})$ ;
- 5   | Atualizar  $\eta$  se  $\mathbf{u}_{k+1}$  for factível e  $G_0^p(\mathbf{u}_{k+1}) < \eta$  ;
- 6   | Atualizar os dados de treinamento  $(\mathbf{U}_{k+1}, \mathbf{y}_{k+1})$  com as medidas em  $\mathbf{u}_{k+1}$  ;
- 7   | Ajustar os GPs sobre o conjunto de dados atualizado
- 8 **até atingir critério de convergência;**

---

## 4.4 Desenho do Experimento

O desempenho de sistemas RTO será comparado para os indicadores propostos por (QUELHAS; JESUS; PINTO, 2013), os quais medem a diferença relativa das variáveis de controle  $\mathbf{u}$  e função objetivo  $G_0$  de cada iteração do RTO em relação aos valores ótimos da planta  $\mathbf{u}^*$  e  $G_0^*$ , conforme definido pelas Equações 4.7 e 4.8. É prática comum na literatura a definição de um modelo representando a planta, o qual é necessário para gerar as amostras do processo requeridas pelo sistema RTO. Portanto, os valores ótimos da planta ( $\mathbf{u}^*$  e  $G_0^*$ ) podem ser facilmente definidos através da solução do Problema 2.1.

$$\Delta \mathbf{u} \% = 100 \left\| \frac{\mathbf{u} - \mathbf{u}^*}{\mathbf{u}^*} \right\| \quad (4.7)$$

$$\Delta G_0(\mathbf{u}) \% = 100 \frac{G_0^p(\mathbf{u}) - G_0^*}{G_0^*} \quad (4.8)$$

O experimento proposto avalia o efeito da presença de ruídos de medição, pontos iniciais e algoritmo de otimização. Os parâmetros restantes do sistema, como a configuração

dos GPs, terão seus valores estimados de forma empírica ou seguindo recomendações da literatura, sendo mantidos fixos durante todas as execuções. Os otimizadores foram utilizados com os parâmetros padrão disponibilizados pelas implementações, simulando o cenário mais simples de sua aplicação, de forma a também permitir uma comparação mais justa. Entretanto, em problemas reais a escolha de um otimizador com parâmetros adequados será uma etapa essencial para obter bons resultados, mas não será explorada neste trabalho devido à simplicidade dos modelos empregados, ficando como sugestão de continuidade. A rotina desenvolvida é apresentada no [Algoritmo 4](#), a qual incorpora de forma genérica a execução dos métodos MA-GP e MA-GP-EIC em sua definição.

Devido à natureza estocástica do processo serão realizadas  $n_R$  repetições da execução de no máximo  $n_C$  iterações do sistema RTO (ciclos), o qual é o critério de parada considerado. Isso permite a estimativa de intervalos de confiança para comparação visual da média dos indicadores ao longo das iterações. Inicialmente são gerados  $n_R$  datasets iniciais distintos de  $n_A$  amostras com ruído Gaussiano aditivo de média zero e variância  $\sigma_r^2$ . Para cada iteração o respectivo dataset é carregado e um ciclo RTO é executado para cada sistema RTO (MA-GP ou MA-GP-EIC) sobre um determinado algoritmo de otimização. Os otimizadores considerados são o DE, SQP e NM, descritos na [Seção 3.2](#). O DE e SQP serão utilizados por todos os sistemas RTO, uma vez que são capazes de lidar com problemas restritos. O NM é executado exclusivamente com o MA-GP-EIC, já que este é um problema irrestrito.

Cada iteração do sistema é sempre inicializada a partir do último ponto de operação avaliado sobre a planta. Antes de ser aplicado no processo, aplica-se um filtro exponencial de primeira-ordem ([Equação 4.2](#)) à solução dos [Problemas 4.1 e 4.6](#). Caso algum algoritmo não encontre uma solução factível para estes problemas, um novo ponto é amostrado em torno do último ponto de iteração para tentar prevenir a convergência do sistema RTO a uma solução sub-ótima. Conforme sugerido por ([FERREIRA et al., 2018](#)), somente as últimas  $n_K$  amostras são consideradas para o treinamento dos GPs. Para prevenir o fenômeno de sobre-ajuste, novos dados são somente incorporados para o treinamento se estiverem suficientemente distantes dos existentes. Para isso, calcula-se a distância Euclidiana em relação às instâncias de treinamento, incorporando-o somente se todas são superiores a um limite  $\Delta_u$ .

#### 4.4.1 Comparação dos intervalos de confiança

Intervalos de confiança (IC) quantificam, para uma significância  $\alpha$ , o grau de incerteza associada à estimativa dos parâmetros de uma população, como sua média e variância ([MONTGOMERY; RUNGER, 2010](#)). A técnica *bootstrapping* ([HAUKOOS; LEWIS, 2005](#)) será utilizada para a estimativa do IC da média em cada iteração, uma vez que ela é não-paramétrica e a única premissa para a sua aplicação é termos uma

**Algoritmo 4:** Experimento completo

---

**Entrada:** Sistemas RTO  $\mathcal{R}$ , Otimizadores  $\mathcal{O}$ ,  $\sigma_r^2$ ,  $n_R$ ,  $n_C$ ,  $n_A$ ,  $\Delta_u$ ,  $n_K$

- 1 Gerar  $n_R$  datasets iniciais distintos de  $n_A$  amostras com ruído de média 0 e variância  $\sigma_r^2$  ;
- 2 Gravar melhor ponto de operação do dataset inicial ;
- 3  $i \leftarrow 0$ ;
- 4 **enquanto**  $i < n_R$  **faça**
- 5     Carregar o dataset inicial para a repetição  $i$  ;
- 6     **para cada**  $RTO \in \mathcal{R}$  **faça**
- 7         **para cada**  $Otim \in \mathcal{O}$  **faça**
- 8              $k \leftarrow 0$ ;
- 9             **enquanto**  $k < n_C$  **faça**
- 10                 Encontrar próximo ponto de operação com  $Otim$  para o problema de otimização do  $RTO$  ;
- 11                 **se** *Solução é infactível* **então**
- 12                     | Amostrar uma nova solução viável aleatoriamente ;
- 13                 **senão**
- 14                     | Filtrar a solução encontrada (Equação 4.2) ;
- 15                 **fim**
- 16                 Amostrar a planta para a solução encontrada ;
- 17                 Adicionar ruído de média 0 e variância  $\sigma_r^2$  à medição ;
- 18                 **se** *Solução atual é factível e melhor que a anterior* **então**
- 19                     | Atualizar melhor ponto de operação com o atual ;
- 20                 **se** *Distância da amostra ao dataset*  $> \Delta_u$  **então**
- 21                     | Adicionar a nova amostra ao dataset ;
- 22                 Adaptar os GPs do  $RTO$  com as últimas  $n_K$  amostras do dataset ;
- 23                  $k \leftarrow k + 1$
- 24             **fim**
- 25     **fim**
- 26     **fim**
- 27      $i \leftarrow i + 1$
- 28 **fim**

**Resultado:** Dados do experimento

---

amostra aleatória (i.i.d.). Em um sistema RTO, espera-se uma maior variabilidade nas iterações iniciais devido ao maior grau de incerteza associado, logo é difícil assumir que a distribuição será idêntica em todas as iterações. Portanto, o uso desta técnica permite a realização da mesma abordagem de avaliação em todas as iterações e para todos os indicadores.

Dada uma amostra aleatória de tamanho  $n$ , o funcionamento do *bootstrap* é bem simples e pode ser resumido pelos seguintes passos:

1. Reamostrar  $n$  amostras com reposição do conjunto original
2. Calcular a média dos dados reamostrados

3. Repetir os itens anteriores  $k$  vezes
4. Estimar o IC para as  $k$  observações da média para uma significância  $\alpha$

Uma recomendação geral é realizar pelo menos  $k = 1000$  reamostragens (HAUKOOS; LEWIS, 2005), pois pelo Teorema Central do Limite haverá a convergência da média para uma distribuição normal. Logo, é possível estimar o IC com a técnica do percentil (EFRON; TIBSHIRANI, 1994) para a significância desejada. Por exemplo, para  $\alpha = 0.05$ , os limites superior e inferior do IC serão o 97.5° e 2.5° percentis, respectivamente. A comparação visual entre dois grupos pode ser realizada verificando se existe interseção entre os IC calculados, para a qual haverá a evidência de que as médias são distintas caso a interseção não seja observada (CUMMING; FINCH, 2005).

## 4.5 Etapas e desafios da implementação

A implementação do experimento é desafiadora devido à característica multidisciplinar e complexidade de sistemas RTO. Além do conhecimento de técnicas de otimização e regressão não linear, também é necessário o entendimento sobre os modelos de processo utilizados como elemento base deste problema. A seguir são descritos os desafios encontrados para auxiliar na reprodutibilidade deste estudo.

### 4.5.1 Modelos de processo

Realizar a simulação de processos regidos por sistemas equações diferenciais na linguagem *Python* é desafiador em um primeiro contato. Embora já exista a implementação dos métodos numéricos apropriados no pacote *scipy*, uma curva de aprendizado considerável pode ser necessária até o ponto de ter confiança para utilizá-los de forma efetiva. Por exemplo, especialmente para processos contínuos, não está explícito na literatura que a forma de calcular a solução em estado estacionário consiste em simplesmente encontrar as raízes do sistema de equações. Recomenda-se utilizar esta abordagem para qualquer modelo contínuo, devido à sua eficiência.

Outro problema que necessita de certa atenção está relacionado aos parâmetros e constantes dos modelos apresentados na literatura. É comum a apresentação de parâmetros em unidades incorretas ou até sem esta informação. Por exemplo, a maioria dos trabalhos apresentam a temperatura em Celsius durante os resultados, mas não mencionam que taxas de reação requerem que esta seja utilizada em Kelvin. É sugerido procurar nas referências se existem os códigos da implementação do modelo disponível para validar novas implementações. Poucas publicações possuem os códigos dos modelos disponibilizados, sendo infelizmente difícil encontrá-los.

## 4.5.2 Otimizadores

Uma recomendação geral para esta etapa é utilizar pacotes (*solvers*) com implementações confiáveis dos algoritmos de otimização a serem utilizados durante o estudo. Além de serem mais eficientes computacionalmente, isso elimina possíveis efeitos indesejados nos resultados decorrentes de erros na implementação dos algoritmos. O pacote *scipy* é suficiente para obter resultados satisfatórios com problemas de baixa escala.

## 4.5.3 Processos Gaussianos

Assim como os otimizadores, recomenda-se o uso de pacotes já validados pela comunidade, como o *scikit-learn*, para realizar o treinamento e predição com GPs. Embora seja relativamente simples implementar os GPs, existe a necessidade do cálculo da inversão de matrizes durante as predições, o que pode inadvertidamente resultar em problemas numéricos e comprometer a utilização do modelo. Além disso, será necessário implementar também a rotina para ajuste dos hiper-parâmetros, o que pode acabar não sendo uma tarefa trivial.

## 4.5.4 Sistema RTO

Esta etapa consiste na integração das partes anteriores para transformação em um sistema RTO completo, o que envolve diversos detalhes de implementação omitidos da literatura. Para simplificá-la, recomenda-se que os itens anteriores sejam implementados de forma modular e com a possibilidade de serem testados individualmente. Este último ponto é muito importante, pois testar o sistema completo é dispendioso e possíveis erros serão mais difíceis de serem encontrados em uma implementação monolítica.

## 4.5.5 Experimento

Implementar o [Algoritmo 4](#) é bem direto, assumindo que o sistema RTO completo foi implementado com sucesso. O desafio real desta etapa é do ponto de vista de coleta de dados experimentais, um vez que é importante gravar de forma eficiente as informações geradas para garantir a veracidade e reprodutibilidade dos resultados. Devido ao tamanho do problema como um todo, uma boa organização é essencial para conseguir executar os experimentos de forma consistente e avaliar todos os aspectos do problema de forma simples.

Para atingir este objetivo, um modelo de dados para o experimento foi desenvolvido onde todas as informações relevantes são salvas em um banco de dados *SQLite* ([HIPP, 2020](#)). Através dele é possível obter desde os valores de função objetivo até os modelos GPs treinados em cada iteração do sistema RTO. A grande vantagem desta abordagem é

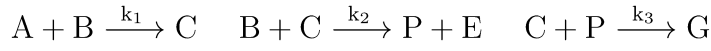
permitir que a geração e análise dos dados sejam realizadas de forma independente, o que garante uma maior agilidade.

## 4.6 Modelos de Processo

Para os modelos descritos a seguir e utilizados neste trabalho, o integrador utilizado para a solução dos sistemas de equações diferenciais é o RK45 (*Explicit Runge-Kutta method of order 5(4)*) (DORMAND; PRINCE, 1980) com uma tolerância absoluta de  $10^{-6}$ , conforme descrito na documentação do pacote *scipy* (VIRTANEN et al., 2020).

### 4.6.1 O Reator de Williams-Otto (RWO)

O reator de Williams-Otto (WILLIAMS; OTTO, 1960) é um processo que opera de forma contínua, sendo usado como *benchmark* para comparação de sistemas RTO (SINGHAL et al., 2016; FERREIRA et al., 2018; SRINIVASAN; BONVIN, 2019). Ele é alimentado por dois reagentes ( $F_A$  e  $F_B$ ), e sua saída é composta por seis componentes envolvidos nas reações:



A espécie  $P$  é o produto desejado,  $E$  um sub-produto com valor de venda,  $C$  um complexo intermediário sem valor agregado, e  $G$  um material residual. As equações de balanço de massa em estado-estacionário podem ser definidas como (ZHANG; FORBES, 2000):

$$F_A - F_R X_a - k_1 V_R X_a X_b = 0 \quad (4.9)$$

$$F_B - F_R X_b - k_1 V_R X_a X_b - k_2 V_R X_b X_c = 0 \quad (4.10)$$

$$2k_1 V_R X_a X_b - F_R X_c - 2k_2 V_R X_b X_c - k_3 V_R X_c X_p = 0 \quad (4.11)$$

$$2k_2 V_R X_b X_c - F_R X_e = 0 \quad (4.12)$$

$$1.5k_3 V_R X_c X_p - F_R X_g = 0 \quad (4.13)$$

$$k_2 V_R X_b X_c - F_R X_p - 0.5k_3 V_R X_c X_p = 0 \quad (4.14)$$

onde  $X_i$  é a concentração de cada espécie  $i \in \{a, b, c, e, g, p\}$  (expresso como frações de massa (WILLIAMS; OTTO, 1960)),  $V_r$  o volume do reator (representado como quilos de líquido para uma densidade de  $3.1214 \text{ kg.m}^{-3}$ , conforme descrito por (WILLIAMS; OTTO, 1960)) e  $F_R = F_A + F_B$  ( $\text{kg.s}^{-1}$ ) a soma da vazão volumétrica dos reagentes. As taxas de



reação  $k_i$  ( $s^{-1}$ ) são definidas como uma função da temperatura do reator  $T_r$  ( $^{\circ}C$ ):

$$k_1 = 1.6599 \times 10^6 e^{\left(\frac{-6666.7}{T_r+273}\right)} \quad (4.15)$$

$$k_2 = 7.2177 \times 10^8 e^{\left(\frac{-8333.3}{T_r+273}\right)} \quad (4.16)$$

$$k_3 = 2.6745 \times 10^{12} e^{\left(\frac{-11111}{T_r+273}\right)} \quad (4.17)$$

As variáveis de decisão deste processo são  $F_B$  e  $T_r$ , enquanto  $F_A$  e  $V_R$  são mantidas em valores constantes de  $1.8257 \text{ kg}\cdot\text{s}^{-1}$  e  $2105.2 \text{ kg}$  respectivamente. Para simular a presença de discrepância estrutural em relação à planta, uma versão simplificada do modelo descrito anteriormente com somente duas equações é considerado:



Isso em resulta nas novas constantes cinéticas:

$$k_1 = 1.655 \times 10^8 e^{\left(\frac{-8077.6}{T_r+273}\right)} \quad (4.18)$$

$$k_2 = 2.611 \times 10^{13} e^{\left(\frac{-12438.5}{T_r+273}\right)} \quad (4.19)$$

e o seguinte sistema de equações de balanço de massa em estado-estacionário (ZHANG; FORBES, 2000):

$$F_A - F_R X_a - k_1 V_R X_a X_b^2 - k_2 V_R X_a X_b X_p = 0 \quad (4.20)$$

$$F_B - F_R X_b - 2k_1 V_R X_a X_b^2 - k_2 V_R X_a X_b X_p = 0 \quad (4.21)$$

$$2k_1 V_R X_a X_b^2 - F_R X_e = 0 \quad (4.22)$$

$$3k_2 V_R X_a X_b X_p - F_R X_g = 0 \quad (4.23)$$

$$k_1 V_R X_a X_b^2 - F_R X_p - k_2 V_R X_a X_b X_p = 0 \quad (4.24)$$

O problema de otimização consiste em encontrar a combinação das variáveis  $F_B$  e  $T_r$  que maximizam o lucro respeitando limites máximos para as espécies A e G quando o sistema está em estado-estacionário:

$$\underset{F_B, T_r}{\text{maximizar}} \quad 1043.38 X_p F_R + 20.92 X_e F_R - 79.23 F_A - 118.34 F_B$$

sujeito a: Modelo (4.20-4.24)

$$X_a \leq 0.12 \quad (4.25)$$

$$X_g \leq 0.08$$

$$F_B \in [3, 6]$$

$$T_r \in [70, 100]$$

Realizando a solução do Problema 4.25 para o modelo (4.9-4.14) com o algoritmo DE, encontra-se como uma solução ótima da planta  $G_0^* = 75.8017$  em  $F_B^* = 4.3898 \text{ kg}\cdot\text{s}^{-1}$  e  $T_r^* = 80.6438 \text{ }^{\circ}C$ . Sua otimalidade está de acordo com os resultados reportados na literatura, conforme pode ser visto em (CHANONA et al., 2019; RIO-CHANONA et al., 2020), por exemplo.

### 4.6.2 O Reator de Acetoacetilação de Pirrol (RAP)

O reator de acetoacetilação de pirrol foi proposto em (RUPPEN; BONVIN; RIPPIN, 1998). Ele opera em regime de semi-batelada e possui as seguintes reações:



Um modelo para este processo pode ser descrito pelo seguinte sistema de equações diferenciais de primeira-ordem (RUPPEN; BONVIN; RIPPIN, 1998):

$$\frac{dc_A}{dt} = -k_1 c_A c_B - \frac{F}{V} c_A \quad (4.26)$$

$$\frac{dc_B}{dt} = -k_1 c_A c_B - 2k_2 c_B^2 - k_3 c_B - k_4 c_B c_C + \frac{F}{V} (c_B^{in} - c_B) \quad (4.27)$$

$$\frac{dc_C}{dt} = k_1 c_A c_B - k_4 c_B c_C - \frac{F}{V} c_C \quad (4.28)$$

$$\frac{dc_D}{dt} = k_2 c_B^2 - \frac{F}{V} c_D \quad (4.29)$$

$$\frac{dV}{dt} = F \quad (4.30)$$

onde  $c_A$ ,  $c_B$ ,  $c_C$  e  $c_D$  são as concentrações de cada espécie,  $V$  é o volume do reator e  $F$  é a vazão volumétrica de alimentação da espécie B com concentração  $c_B^{in}$ , e  $k_1$ ,  $k_2$ ,  $k_3$  e  $k_4$  são os coeficientes cinéticos de cada reação. Os valores de cada parâmetro e condições iniciais estão apresentados na Tabela 1.

Tabela 1 – Valores dos parâmetros e condições iniciais. Coeficientes cinéticos  $k_1$ ,  $k_2$ ,  $k_4$  estão em  $L \cdot mol^{-1} \cdot min^{-1}$  e  $k_3$  em  $min^{-1}$ , concentrações das espécies em  $mol \cdot L^{-1}$  e volume em litros

Parâmetro	Valor	Condições iniciais	Valor
$k_1$	0.053	$c_A(0)$	0.72
$k_2$	0.128	$c_B(0)$	0.05
$k_3$	0.028	$c_C(0)$	0.08
$k_4$	0.001	$c_D(0)$	0.01
$c_B^{in}$	5	$V(0)$	1

Este mecanismo completo de reação será considerado como a planta, enquanto que um modelo aproximado será construído considerando somente as duas primeiras reações, de forma que as demais sejam desconhecidas ( $k_3 = k_4 = 0$ ). O problema de otimização consiste em determinar o perfil de alimentação da espécie B que maximiza a quantidade de C no final do processo, mantendo as concentrações finais de B e C abaixo dos limites especificados:

$$\begin{aligned}
& \underset{u(t)}{\text{maximizar}} && c_C(t_f)V(t_f) \\
& \text{sujeito a:} && \text{Modelo (4.26-4.30)} \\
& && c_B(t_f) \leq c_B^{max} \\
& && c_D(t_f) \leq c_D^{max} \\
& && 0 \leq u(t) \leq F^{max}
\end{aligned} \tag{4.31}$$

com  $t_f = 250 \text{ min}$ ,  $F^{max} = 0.002 \text{ L.min}^{-1}$ ,  $c_B^{max} = 0.025 \text{ mol.L}^{-1}$  e  $c_D^{max} = 0.15 \text{ mol.L}^{-1}$ . Conforme descrito em (CHACHUAT; SRINIVASAN; BONVIN, 2009), originalmente este é um problema de otimização dinâmica, o qual é parametrizado para permitir sua solução como um problema estático. O perfil de alimentação  $u(t)$  será dividido em três partes e descrito por meio de três variáveis  $(t_m, F_s, t_s)$ , onde  $t_m$  é o instante de troca entre a primeira e segunda parte,  $F_s$  é a vazão constante ao longo da segunda parte e  $t_s$  o tempo de troca entre a segunda e terceira partes. Na primeira e terceira partes as vazões são constantes em  $F^{max}$  e 0, respectivamente. Realizando a solução do Problema 4.31 com o algoritmo DE, encontra-se como uma solução ótima  $G_0^* = 0.5085$  e  $\mathbf{u}^* = [18.6139, 0.0011, 227.6375]$ . Sua otimalidade global pode ser verificada comparando com os resultados reportados em (CHACHUAT; SRINIVASAN; BONVIN, 2009; NAVIA et al., 2015), por exemplo.

## 5 Resultados

### 5.1 Introdução

Neste capítulo são apresentados os resultados da execução do experimento proposto no [Capítulo 4](#). Eles serão discutidos para se avaliar o atingimento dos objetivos principais deste trabalho. Primeiramente são apresentados resultados relativos à validação das implementações, seguido da análise dos otimizadores para a abordagem restrita MA-GP e irrestrita MA-GP-EIC, respectivamente. Por fim, uma comparação dos resultados nas duas abordagens MA-GP e MA-GP-EIC é realizada.

### 5.2 Recursos computacionais

Os experimentos foram realizados em um notebook equipado com processador Intel Core i7 de 2.40Ghz (8 CPUs) e 8Gb de memória RAM. O sistema operacional é o Windows 10, utilizando a versão 3.9 da linguagem *Python*. Para ter um melhor desempenho, as rotinas foram executadas dentro uma máquina virtual Linux Ubuntu 18.04 disponível através do recurso WSL. Os algoritmos de otimização utilizados estão disponíveis no pacote *scipy* ([VIRTANEN et al., 2020](#)) e os GPs são treinados com o pacote *scikit-learn* ([PEDREGOSA et al., 2011](#)). O código desenvolvido está disponível no seguinte repositório Git: <https://github.com/vicrsp/rto>.

### 5.3 Dados Experimentais

Um total  $n_R = 30$  repetições foram executadas para cada experimento completo, com ciclos RTO de tamanho  $n_C^{RWO} = 30$  e  $n_C^{RAP} = 60$ , que são suficientes para a convergência do sistema. Estes valores foram determinados de forma empírica, utilizando a velocidade de convergência observada para cada modelo como referência. Para inicialização do sistema RTO, 5 pontos iniciais viáveis são obtidos através de uma amostragem aleatória uniforme para um intervalo de  $\pm 20\%$  em torno de uma solução inicial viável sub-ótima. Para o RWO foi considerada a solução ótima do modelo aproximado  $\mathbf{u}_0^{RWO} = [4.7889, 83.2043]$ . No caso do RAP, foi utilizado o ponto de operação  $\mathbf{u}_0^{RAP} = [10.6521, 0.0005, 224.4806]$ .

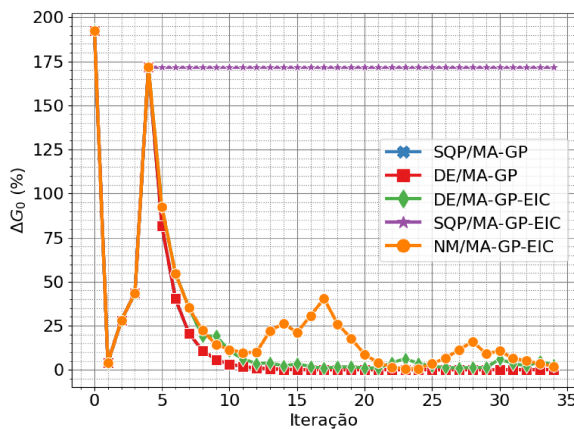
Exceto na etapa de validação, as amostras da função de custo e restrições foram corrompidas com ruído Gaussiano aditivo de média zero e variância 1%. Um máximo de 1000 iterações ou uma tolerância absoluta de  $10^{-6}$  para a variação da função objetivo foram utilizados como critérios de paradas para os otimizadores DE, SQP e NM. Estes valores foram escolhidos para garantir que os algoritmos tenham tempo suficiente para

encontrar a melhor solução possível. Os demais hiper-parâmetros foram mantidos fixos com os valores padrão utilizados no pacote *scipy*.

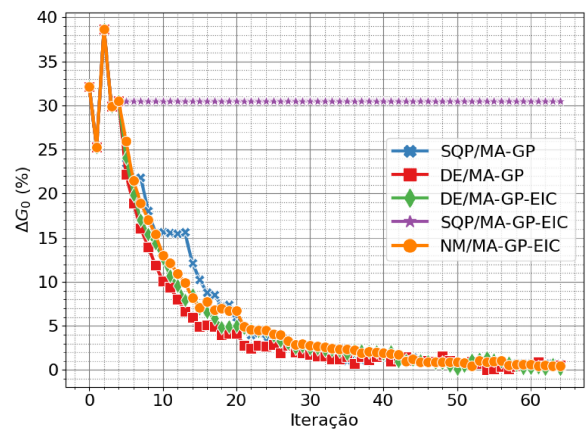
Os GPs consideram um *kernel* RBF com função de média  $c = 0$ , covariância  $\sigma_n^2 = 1$  e  $\Lambda = \mathbf{I}$ , o qual é sempre atualizado sobre as últimas  $n_K = 10$  amostras em cada iteração do ciclo RTO. Antes da inferência foi necessário padronizar os dados de entrada e saída através da subtração da média e divisão pelo desvio padrão, evitando assim problemas numéricos devido a variáveis em escalas diferentes. Para prevenir o fenômeno de sobre-ajuste durante o treinamento, é considerado um limite  $\Delta_{\mathbf{u}} = 0.01$ . Este valor foi definido de forma empírica, uma vez que não foi encontrada uma recomendação específica na literatura. As soluções ótimas encontradas para o Problema 2.2 são filtradas por uma matriz de ganho constante  $\mathbf{K} = 0.4\mathbf{I}$  (Equação 4.2), o que resulta em uma convergência mais fácil (FERREIRA et al., 2018).

## 5.4 Validação da implementação

A validação da implementação é feita sobre um cenário sem ruídos de medição e somente com uma repetição do experimento. Para cada modelo de processo, os resultados da diferença relativa da função objetivo estão disponíveis na Figura 6. Nela são apresentadas todas as soluções avaliadas sobre a planta em cada iteração, seguindo o mesmo padrão da literatura. Comparando com os resultados de (CHACHUAT; SRINIVASAN; BONVIN, 2009; FERREIRA et al., 2018), por exemplo, é possível constatar que a implementação foi realizada com sucesso. As primeiras 5 iterações são idênticas, evidenciando que todos são inicializados nas mesmas condições experimentais.



(a) RWO



(b) RAP

Figura 6 – Diferença relativa da função objetivo para as execuções de validação

Um resultado interessante pode ser observado para o MA-GP-EIC com SQP, no qual o sistema é incapaz de melhorar o ponto de operação em que é inicializado para os dois

modelos. A justificativa deste comportamento está relacionada com a superfície de decisão da função EIC, a qual pode ser vista na [Figura 7](#). Observe que ela é predominantemente plana, logo o SQP ficará preso se inicializado em qualquer ponto deste platô. Isso ocorre pois ele depende da informação dos gradientes de primeira ordem e da Hessiana, os quais por serem nulos causam a sua convergência prematura ([KRAFT et al., 1988](#)).

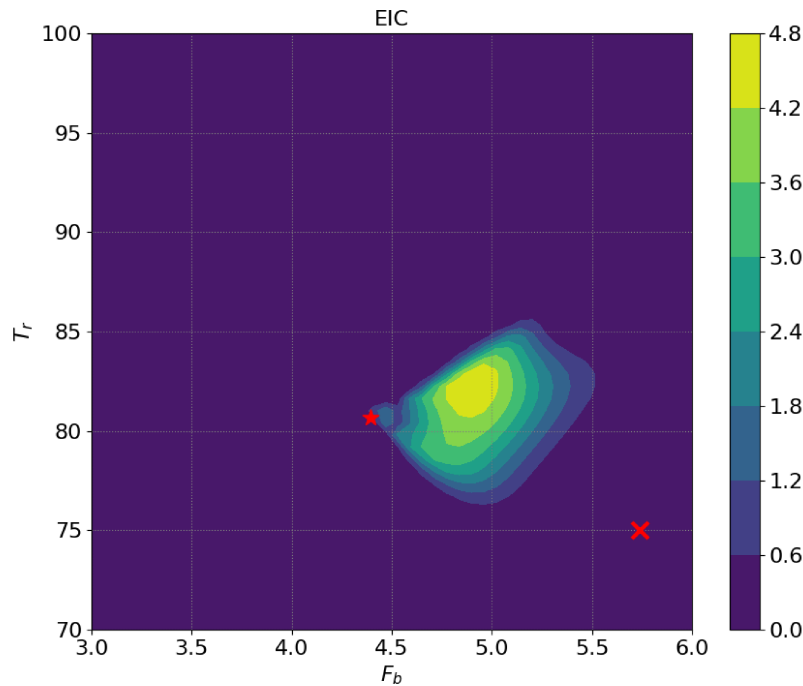


Figura 7 – Superfície de decisão do MA-GP-EIC com SQP para o RWO.  $\times$  representa a solução final encontrada pelo algoritmo SQP e  $\star$  o ótimo da planta.

## 5.5 Escolha do algoritmo de otimização

Conforme visto anteriormente, alguns otimizadores já apresentam dificuldades dependendo da abordagem RTO e modelo de processo, em especial o SQP aplicado no MA-GP-EIC. Optou-se portanto por descartá-lo dos demais experimentos pois o mesmo comportamento é esperado. Uma sugestão para lidar com esse problema é a aplicação de uma heurística de reinicialização do algoritmo, conforme sugerido em ([RIO-CHANONA et al., 2020](#)). Ela não foi explorada neste trabalho, ficando como possível sugestão de continuidade. A seguir, é feita uma análise para definir os otimizadores mais adequados para cada abordagem RTO. Para todos os gráficos a região sombreada é o intervalo de confiança de 95% para a média obtido com a técnica de *bootstrapping*. Além disso, a média é calculada para todas as soluções avaliadas na planta em um determinada iteração, que é o formato mais comum na literatura.

### 5.5.1 MA-GP

Através da [Figura 8](#) é possível comparar os resultados da diferença relativa da função objetivo para cada problema benchmark com o MA-GP. É possível notar que para o RWO, os algoritmos DE e SQP obtiveram perfis de convergência muito similares, para o qual podemos concluir que ambos são adequados para este problema. Entretanto, o mesmo comportamento não é exibido para o RAP: note que além de apresentar uma variabilidade muito maior ao longo das iterações, o SQP converge em média para um ponto de operação sub-ótimo da planta. Esta variabilidade está relacionada com possíveis limitações do SQP para este problema, o qual pode estar ficando preso em ótimos locais durante as iterações.

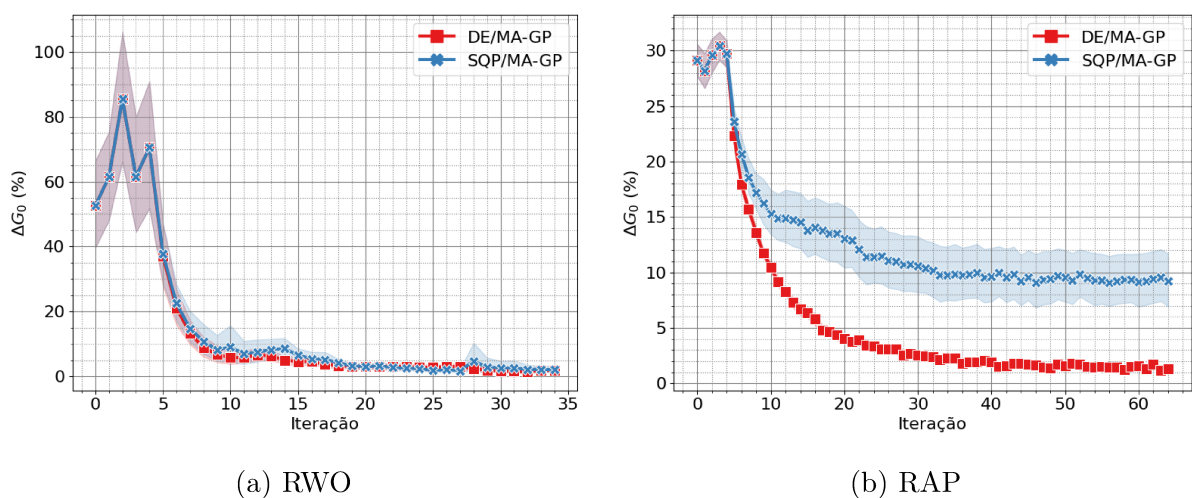


Figura 8 – Diferença relativa da função objetivo para o MA-GP

Os gráficos *boxplot* do tempo de execução das iterações do sistema RTO para cada otimizador estão disponíveis na [Figura 9](#). Conforme esperado, por ser um heurística que necessita de um número maior de execuções do modelo de processo, o DE possui um custo superior ao SQP, sendo 13x mais lento no RWO e 8x para o RAP. Apesar deste resultado, o DE é escolhido como o melhor algoritmo para o RTO MA-GP. Isso se deve ao fato dele ser menos afetado por problemas numéricos, apresentando convergência superior no problema RAP para um intervalo de confiança de 95%. É importante ressaltar que em problemas reais esse aumento no custo computacional pode ser tornar proibitivo, o que não ocorre nos modelos de *benchmark* usados nesse trabalho.

### 5.5.2 MA-GP-EIC

Os valores de diferença relativa da função objetivo para cada modelo de processo pode ser visto na [Figura 10](#). Através deste resultado, não podemos afirmar que um dos otimizadores foi superior ao outro para um intervalo de confiança da 95%, e o critério de convergência considerado. É importante observar, entretanto, que o NM apresenta uma maior variabilidade nas iterações iniciais para o RWO, além de uma convergência

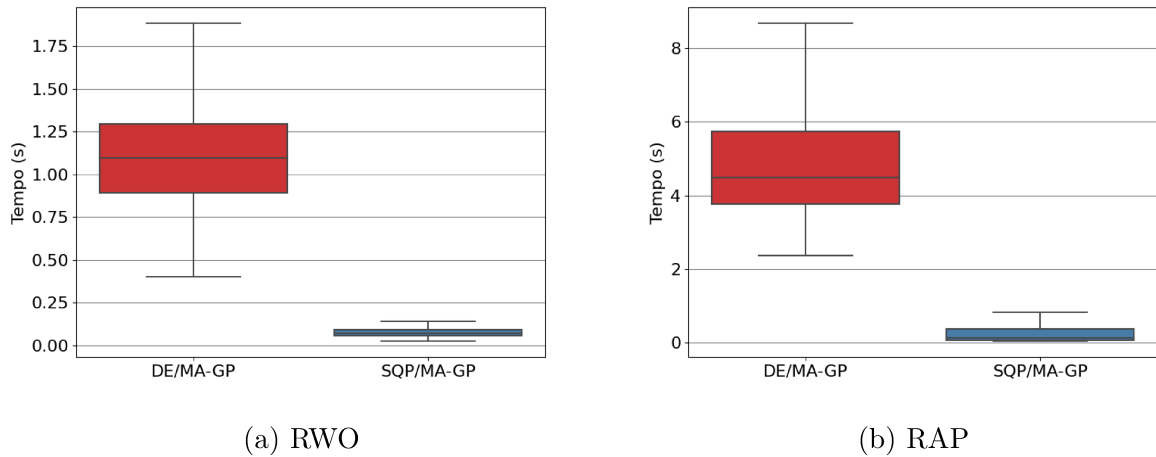


Figura 9 – *Boxplot* do custo computacional das iterações do MA-GP

mais lenta. Isso não foi observado para o RAP, para o qual o NM apresentou uma convergência ligeiramente mais acelerada do que o DE nas iterações finais. Isso sugere que as características do problema RWO podem estar acentuando a não-convexidade da função EIC durante as iterações iniciais, causando a convergência do NM para ótimos locais.

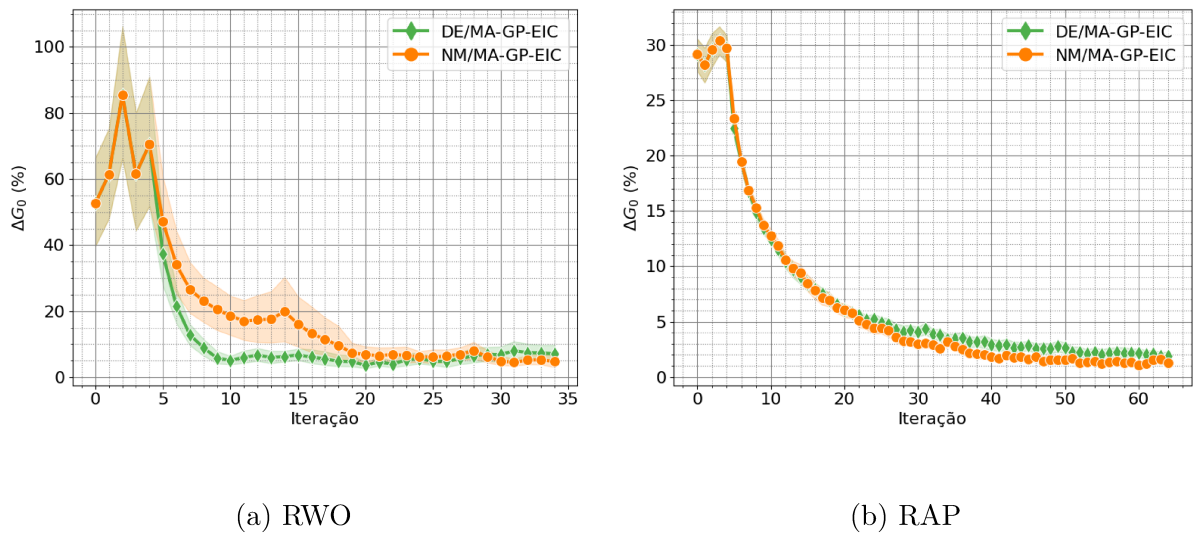


Figura 10 – Diferença relativa da função objetivo para o MA-GP-EIC

Na [Figura 11](#) é exibido o esforço computacional obtido, onde o DE possui um custo mais elevado, sendo este 8x maior no RAP e 7x no RWO. Como o esforço computacional do NM é muito menor, é importante entender o que pode estar causando sua maior variabilidade inicial e menor velocidade de convergência para o RWO. Uma primeira hipótese consiste em verificar o comportamento das variáveis de controle calculadas em cada iteração, onde espera-se que isso cause variabilidade nos demais indicadores. Conforme pode ser visto na [Figura 12](#), esta hipótese inicial é válida, comprovando o efeito da convergência a ótimos locais do NM sobre o desempenho do sistema RTO quando inicializado de diferentes pontos de operação. Para complementar a afirmação anterior, avaliando também



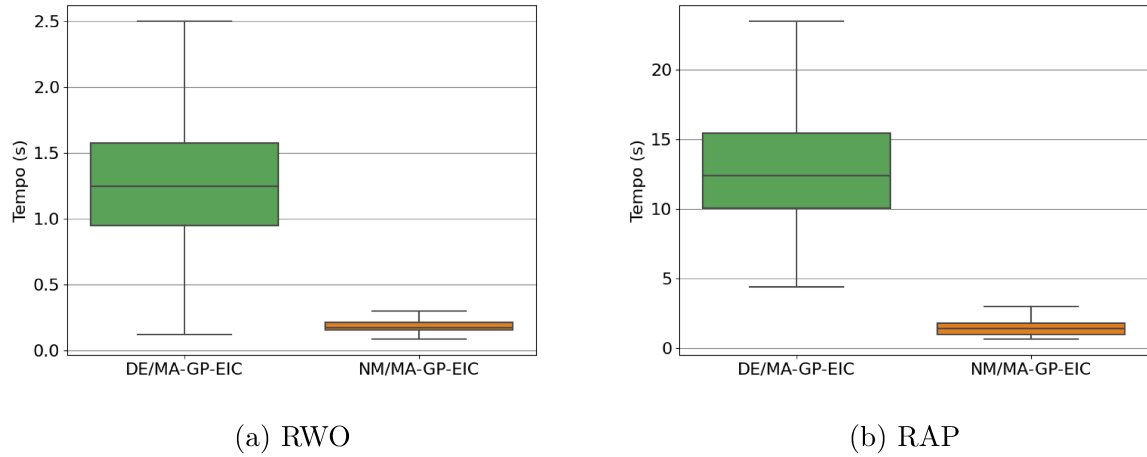


Figura 11 – *Boxplot* do custo computacional das iterações do MA-GP-EIC

as restrições para o RWO (Figura 13) é possível notar que o NM convergiu para ótimos locais durante as iterações iniciais, uma vez que ele segue um caminho que o mantém mais distante da fronteira das restrições. Embora este comportamento seja interessante do ponto de vista prático, isso mostra que o DE é mais adequado pois possui menor sensibilidade à inicialização do sistema RTO.

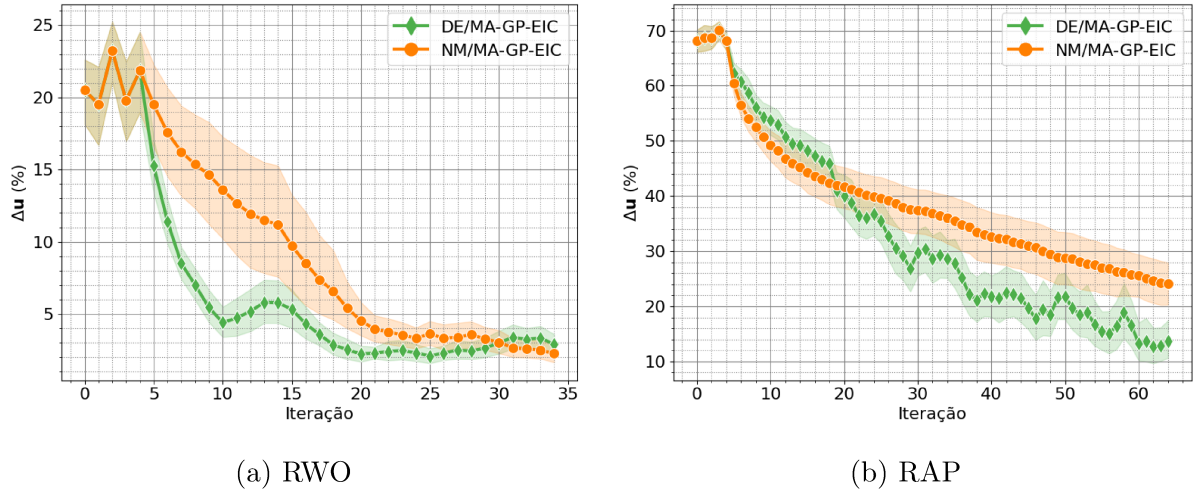


Figura 12 – Diferença relativa dos sinais de controle para o MA-GP-EIC

## 5.6 Comparação entre RTO MA-GP e MA-GP-EIC

Através da discussão anterior, conclui-se que o DE foi adequado para ambas abordagens RTO e modelos de processo, embora resulte em um maior esforço computacional. Portanto, uma primeira análise entre o RTO MA-GP e MA-GP-EIC é realizada para este algoritmo, o qual minimizará o efeito de um otimizador inadequado. Na Figura 14 é possível comparar a convergência entre as duas técnicas com o uso do DE. Considerando o

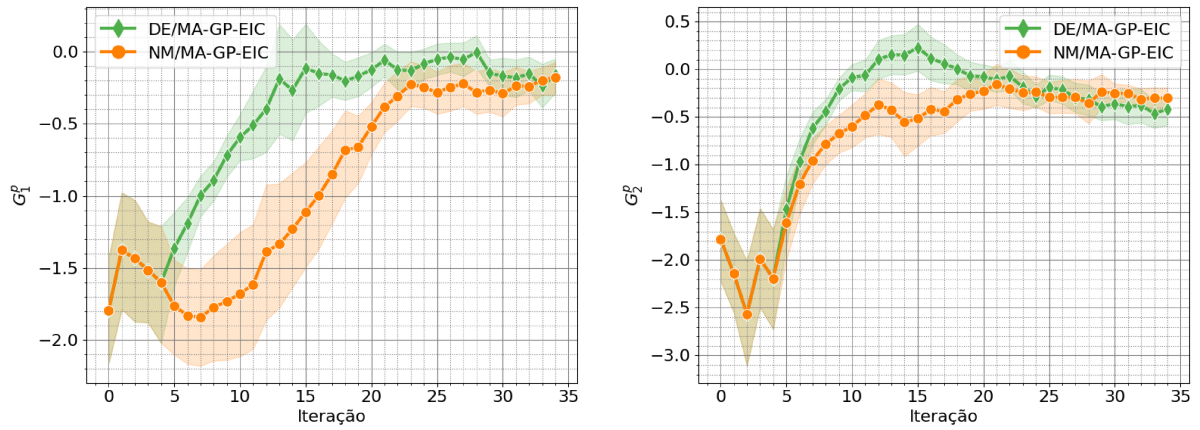


Figura 13 – Restrições do MA-GP-EIC para o problema RWO

intervalo de confiança de 95%, podemos afirmar que o MA-GP teve convergência superior somente para o RWO. Analisando a diferença relativa dos sinais de controle (Figura 15), é possível notar que no RAP o MA-GP-EIC em média está sempre mais distante do ótimo da planta ao longo das iterações do sistema, o que explica a diferença observada para a função objetivo.

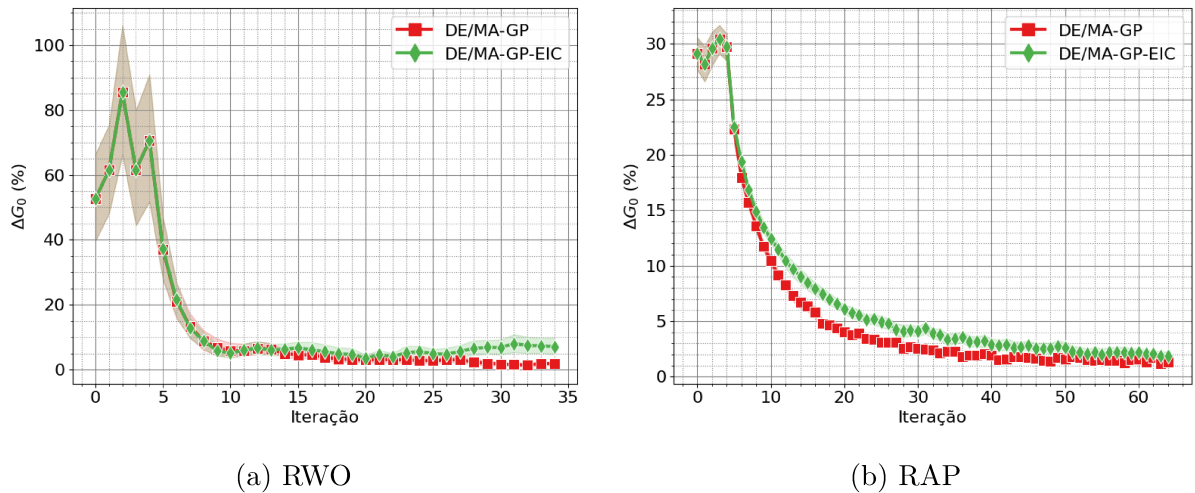


Figura 14 – Diferença relativa da função objetivo para os melhores otimizadores.

Uma hipótese para este resultados está na forma com que cada sistema RTO lida com as restrições do problema. Com o auxílio da Figura 16, é possível notar que o MA-GP-EIC é mais conservador em relação às restrições, ou seja, tem uma maior tendência à respeitá-las em detrimento à convergência. Isso pode ser explicado por duas características da função de aquisição EIC:

- a) O tratamento das restrições é feito de forma similar á técnica de penalidades. Portanto, como nas iterações iniciais a incerteza em relação às restrições é maior, haverá uma tendência de penalizar soluções próximas aos limites de viabilidade.

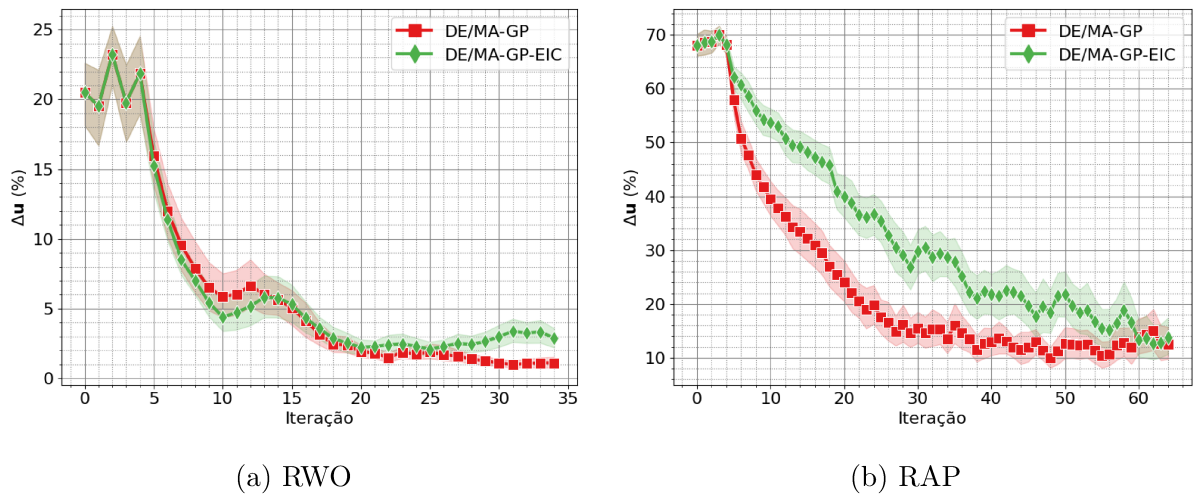


Figura 15 – Diferença relativa dos sinais de controle para os melhores otimizadores.

Dessa forma, o RTO tende a explorar inicialmente soluções mais distantes das fronteiras das restrições.

- b) Durante as iterações do sistema RTO, a EIC irá tentar balancear a exploração e refino das soluções. À medida em que o sistema se aproxima do ótimo da planta, ao mesmo tempo ele se aproxima das restrições. Logo, neste cenário podem ser realizadas iterações de exploração mais afastadas das restrições com o intuito de reduzir as incertezas em relação à função objetivo, a qual poderão ser sub-ótimas.

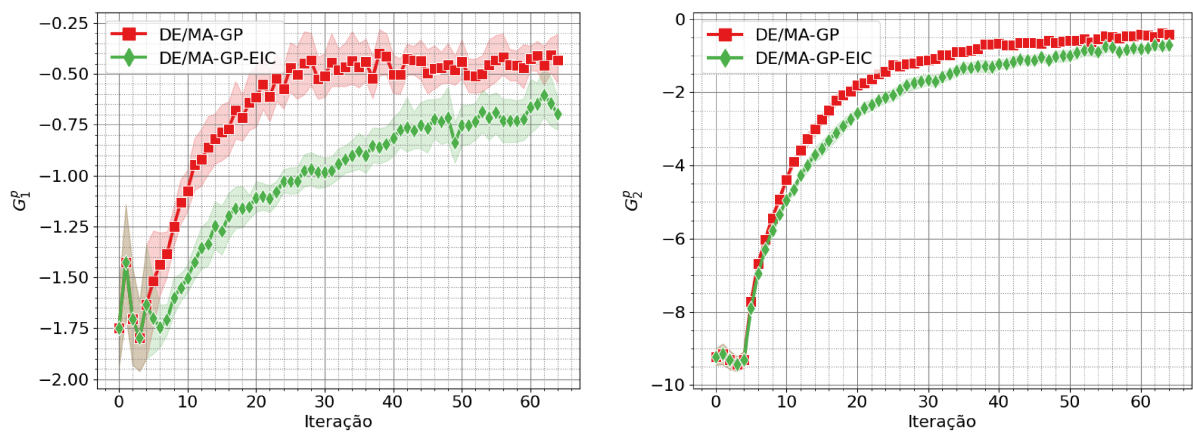


Figura 16 – Restrições para o problema RAP

O comportamento descrito no [Item b\)](#) é melhor observado para o RWO, uma vez que o ótimo da planta ocorre exatamente na interseção entre duas restrições. Através da [Figura 17](#) é possível observar que nas últimas iterações existe uma tendência de exploração da região viável, uma vez que já existe conhecimento suficiente das restrições. Isto causou o deslocamento em relação ao ótimo da planta, ou seja, o uso da função de aquisição EIC

pode resultar em exploração desnecessária e comprometer a convergência do sistema RTO para o critério de parada utilizado. Observe que o MA-GP-EIC consegue convergir em cerca de 15 iterações para uma solução próxima do ótimo da planta, apresentando pouca evolução no objetivo econômico após esse ponto e sua piora próximo do limite de iterações. Logo, a avaliação de um critério de parada mais adequado pode ser investigado como melhoria para o MA-GP-EIC.

Uma última hipótese está relacionada à falta de um tratamento específico quando um novo ponto de operação avaliado apresenta piora do objetivo econômico ou violação de restrições. Ao invés de utilizar este ponto como referência para o cálculo da próxima iteração, o mesmo poderia ser utilizado somente para treinamento dos GPs, sendo o próximo ponto calculado a partir da melhor solução encontrada até o momento, por exemplo. O descarte da etapa de adaptação deve ser melhor avaliado, uma vez que poderíamos perder informações relevantes sobre o comportamento da planta. Esta modificação poderia evitar as explorações desnecessárias observadas, porém é necessário avaliar sua aplicabilidade prática e também se essa alteração poderia levar à convergência indesejada para ótimos locais.

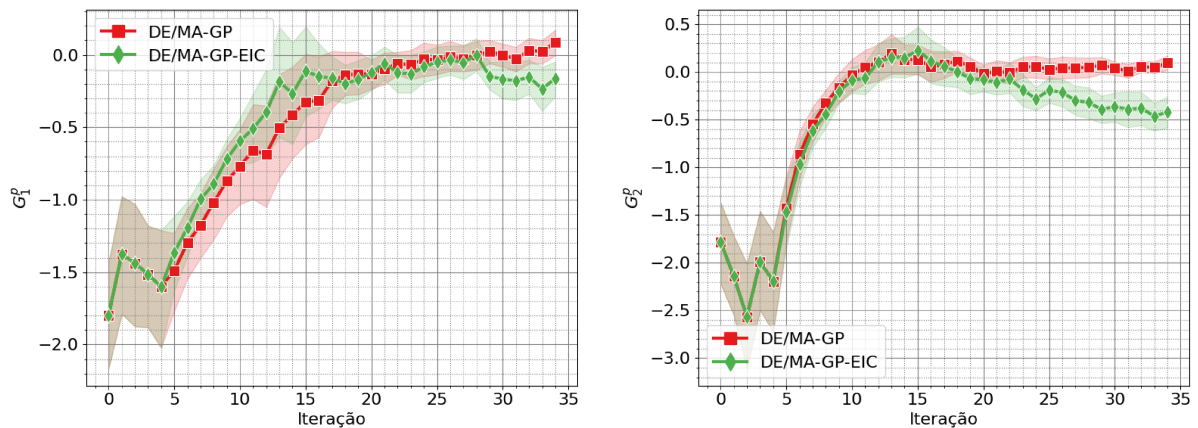


Figura 17 – Restrições para o problema RWO

Através da análise apresentada é possível concluir que o MA-GP-EIC é capaz de levar o sistema ao ótimo da planta, com desempenho não muito distante ao do MA-GP. Ele possui a característica interessante de ser mais conservadora em relação às restrições, o que é um resultado interessante do ponto de vista prático. Entretanto, existe um problema de convergência introduzido pela função de aquisição EIC que causa uma exploração desnecessária com um critério de parada inadequado, especialmente quando o ótimo da planta está na interseção de suas restrições. Logo, é necessário um estudo mais aprofundado visando aprimorar este aspecto indesejado e torná-la mais robusta.

Caso a utilização do DE não seja possível devido a limitações de recursos computacionais, é importante avaliar as duas técnicas quando algoritmos determinísticos são

utilizados. O esforço computacional é apresentado na [Figura 18](#), onde o NM possui um custo cerca de 2x maior que o SQP nos dois problemas. Apesar disso, este valor continua sendo muito inferior em relação ao DE. O MA-GP-EIC apresentou um resultado muito interessante, pois seu desempenho é superior no RAP para um intervalo de confiança de 95%, conforme a diferença relativa da função objetivo apresentada na [Figura 19](#). É importante mencionar sua maior variabilidade inicial no RWO, a qual por não ter sido observada para o SQP pode estar relacionada a uma possível convergência local devido ao uso da função EIC, uma vez que o DE não apresenta este comportamento. Apesar desse resultado, não há evidências suficientes para descartar a equivalência dos resultados para o RWO. Isso mostra que a utilização de uma abordagem de otimização irrestrita permitiu o uso de um algoritmo de busca direta, o qual se mostrou eficiente na solução do Problema 4.6. Dessa forma, o MA-GP-EIC é capaz de reduzir consideravelmente o esforço computacional com o NM, conseguindo ao mesmo tempo mais robustez ao ser aplicado em um problema mais difícil (RAP) e com desempenho próximo ao DE.

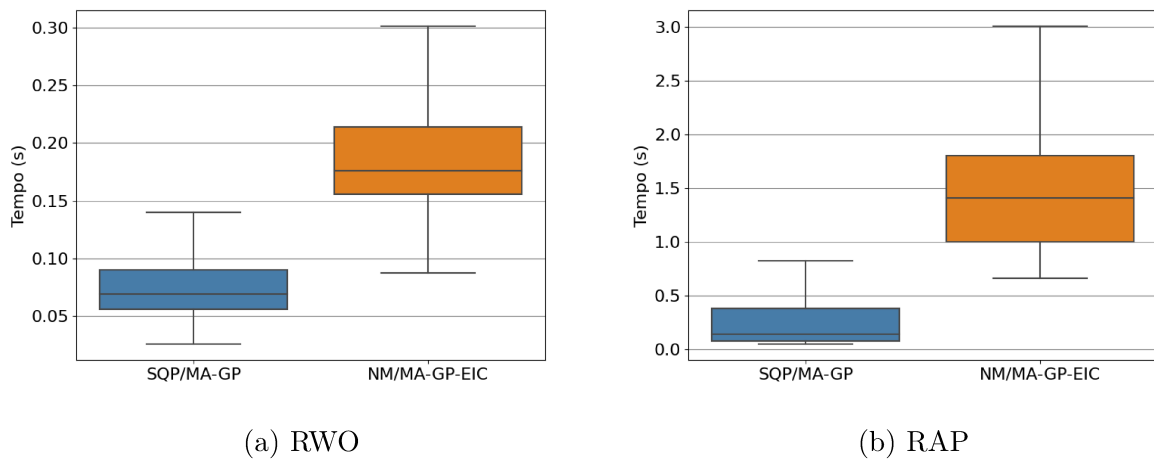
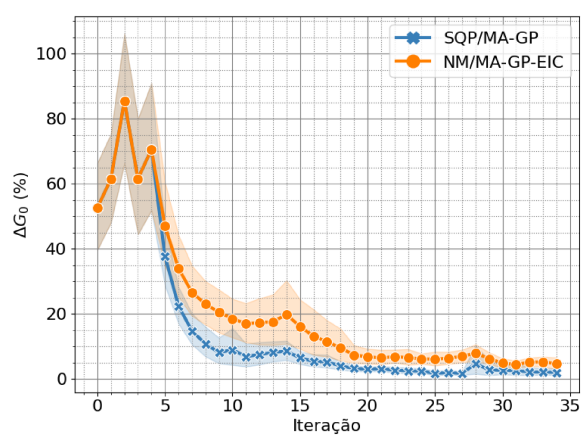
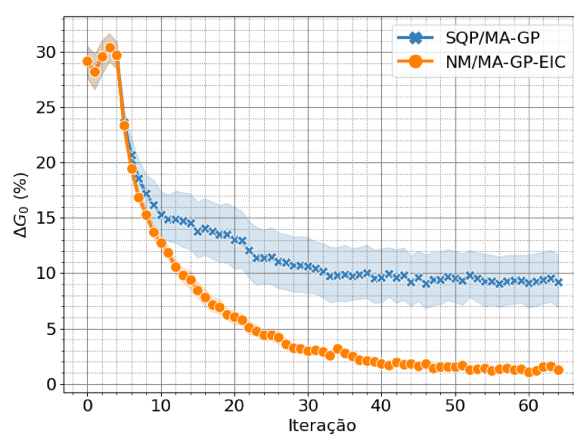


Figura 18 – *Boxplot* do custo computacional do MA-GP com SQP e MA-GP-EIC com NM



(a) RWO



(b) RAP

Figura 19 – Diferença relativa da função objetivo para os otimizadores SQP e NM.

## 6 Conclusão

### 6.1 Considerações Finais

O trabalho desenvolvido consistiu na avaliação do efeito do otimizador no desempenho de sistemas RTO aplicados a modelos de *benchmark*, considerando a presença de ruídos de medição Gaussianos e inicialização em pontos de operação factíveis aleatórios. Para isto, foram considerados as técnicas MA-GP, usualmente utilizada na literatura, e a MA-GP-EIC proposta neste trabalho, executadas com os algoritmos SQP, DE e NM. A primeira etapa consistiu em validar a implementação dos modelos de processo e demais algoritmos desenvolvidos sobre um cenário ideal, onde por comparação com resultados reportados na literatura comprovou-se a sua eficácia. Outro resultado importante desta etapa foi a constatação de que o SQP não é adequado para o MA-GP-EIC, pois a superfície de decisão deste problema é caracterizada por regiões planas.

Na segunda etapa foi realizada a comparação visual dos intervalos de confiança dos experimentos com o intuito de escolher o algoritmo mais adequado para cada abordagem RTO. Para o MA-GP, DE e SQP tiveram desempenhos similares para o problema RWO, entretanto o SQP começou a apresentar problemas numéricos para o RAP. No caso do MA-GP-EIC, não há um otimizador melhor em relação à variação relativa da função objetivo para um intervalo de confiança de 95%, independente do modelo. Entretanto, foi observada uma maior variabilidade do NM durante as iterações iniciais, especialmente para o RWO. Com a análise da variação relativa dos sinais controláveis, comprovou-se sua convergência a ótimos locais. Conforme esperado, o DE necessitou de um maior esforço computacional em relação ao SQP e NM. Dado o baixo custo dos modelos de *benchmark* empregados, optou-se por seguir com o DE como o melhor otimizador para as duas abordagens RTO.

Finalmente, na última etapa foi feita a comparação entre o MA-GP e MA-GP-EIC com o DE. Para o intervalo de confiança de 95%, o MA-GP foi superior em relação à variação relativa da função objetivo, mas somente para o RWO. Entretanto, analisando a variação relativa dos sinais controláveis e as restrições, foram observados dois comportamentos indesejados para MA-GP-EIC que estão comprometendo sua convergência ao ótimo da planta. O primeiro é relacionado à forma como as restrições são tratadas pelo algoritmo, fazendo com que ele tenha preferência a explorar inicialmente soluções interiores à região factível. A segunda é uma característica da função EIC, a qual tende a propor explorações desnecessárias e deslocar o sistema do ótimo da planta. Este efeito ficou mais evidente para o RWO, já que o ótimo da planta se encontra na interseção de suas restrições. Estes dois pontos devem ser investigados com mais detalhes em trabalhos futuros para

um melhor entendimento dos critérios de convergência desta nova abordagem. Caso a disponibilidade de recursos computacionais seja limitada, o MA-GP-EIC com NM tornou-se uma opção muito interessante em relação às demais abordagens. Isso evidencia as vantagens da formulação irrestrita do problema de otimização, a qual necessita de menor esforço computacional e foi mais robusta para um problema mais difícil (RAP). Entretanto, é necessária uma investigação para entender o motivo da convergência mais rápida do MA-GP com SQP no RWO.

## 6.2 Propostas de Continuidade

Neste trabalho foi investigada uma nova abordagem que explora as semelhanças entre o problema RTO e a otimização Bayesiana restrita. Os resultados apresentados para a função de aquisição EIC mostram que o uso destas técnicas são bastante promissoras, principalmente por permitir a solução do problema de otimização baseado em modelos com otimizadores irrestritos. Entretanto, algumas melhorias ainda são necessárias, especialmente em relação à sua convergência. Entre elas, uma primeira consiste na investigação de outras estratégias Bayesianas para problemas restritos, como por exemplo o uso de funções de aquisição restritas baseadas em entropia propostas por (GELBART, 2015). De forma geral, deve-se buscar por uma abordagem que previna explorações desnecessárias quando se acredita estar próximo do ótimo da planta.

Além da inicialização do sistema e presença de ruídos de medição, existem outros fatores que poderiam ser considerados, e que também podem ter impactos visíveis no desempenho de um sistema RTO. Por exemplo, a função de *kernel* e os hiper-parâmetros de treinamento dos GPs podem ter um efeito importante sobre o seu desempenho. Para que este não se torna-se um trabalho extremamente longo e mais complexo, o estudo destes outros fatores fica como sugestão de continuidade. De fato, não existe nenhum trabalho na literatura com o enfoque no aprendizado dos GPs, sendo uma excelente oportunidade para uma possível contribuição relevante.

Outra sugestão é a melhoria da robustez dos algoritmos de otimização empregados. Inicialmente, o uso de uma heurística de *multi-start* para o NM é algo interessante de ser avaliado para melhorar seus resultados. Outra sugestão é o ajuste de hiper-parâmetros do algoritmo DE (HUANG; LI; YAO, 2019) com o objetivo de reduzir seu esforço computacional. Indo mais além, outras meta-heurísticas também poderiam ser avaliadas para tentar melhorar este aspecto. Finalmente, pode-se também estudar o uso de abordagens híbridas envolvendo algoritmos evolutivos e determinísticos, para as quais existem resultados promissores de sua eficácia quando aplicadas a sistemas RTO (QUELHAS; JESUS; PINTO, 2013). Isso poderá ter resultados muito bons para o MA-GP-EIC, especialmente se combinado com o algoritmo NM.



---

Uma última sugestão é a aplicação da metodologia desenvolvida sobre problemas RTO de maior escala. O objetivo sugerido é investigar se os mesmos problemas numéricos serão observados e seu nível de efeito sobre o desempenho do sistema RTO. Isso seria importante para avaliar também se o custo computacional de heurísticas evolutivas seria um fator impeditivo ao abordar problemas com maior escala. Isso vem de encontro à uma deficiência encontrada na literatura, a qual é baseada principalmente em problemas de baixa escala. Por exemplo, a validação do MA-GP sobre um problema complexo fomentaria a possibilidade de sua posterior aplicação em problemas reais, possibilitando novos avanços e aplicações de impacto em problemas práticos que se beneficiariam de uma estratégia de RTO.

# Referências

- ANDERSSON, L. E.; BRADFORD, E. C.; IMSLAND, L. Distributed learning and wind farm optimization with gaussian processes. In: *American Control Conference (ACC), online conference, accepted, 2020a. a, b.* [S.l.: s.n.], 2020. Citado 4 vezes nas páginas 18, 19, 27 e 28.
- ANDERSSON, L. E.; IMSLAND, L. Real-time optimization of wind farms using modifier adaptation and machine learning. *Wind Energy Science*, Copernicus GmbH, v. 5, n. 3, p. 885–896, 2020. Citado na página 29.
- BAGAJEWICZ, M. A brief review of recent developments in data reconciliation and gross error detection/estimation. *Latin American Applied Research*, v. 30, n. 4, p. 335–342, 2000. Citado 2 vezes nas páginas 16 e 23.
- BARD, Y. Nonlinear parameter estimation. Academic press, 1974. Citado na página 23.
- BATHELT, A.; RICKER, N. L.; JELALI, M. Revision of the tennessee eastman process model. *IFAC-PapersOnLine*, Elsevier, v. 48, n. 8, p. 309–314, 2015. Citado na página 28.
- BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. *Nonlinear programming: theory and algorithms.* [S.l.]: John Wiley & Sons, 2013. Citado 3 vezes nas páginas 30, 31 e 32.
- BIEGLER, L. T. Optimization strategies for complex process models. *Advances in Chemical Engineering*, Academic Press, v. 18, p. 197–256, 1992. Citado na página 22.
- BIEGLER, L. T.; ZAVALA, V. M. Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization. *Computers & Chemical Engineering*, Elsevier, v. 33, n. 3, p. 575–582, 2009. Citado na página 15.
- BONVIN, D.; SRINIVASAN, B.; RUPPEN, D. *Dynamic optimization in the batch chemical industry.* [S.l.], 2001. Citado na página 28.
- BRDYŚ, M.; TATJEWSKI, P. An algorithm for steady-state optimizing dual control of uncertain plants. In: *New Trends in Design of Control Systems 1994.* [S.l.]: Elsevier, 1995. p. 215–220. Citado na página 25.
- CÂMARA, M. M.; QUELHAS, A. D.; PINTO, J. C. Performance evaluation of real industrial rto systems. *Processes*, Multidisciplinary Digital Publishing Institute, v. 4, n. 4, p. 44, 2016. Citado na página 22.
- CÂMARA, M. M. et al. Numerical aspects of data reconciliation in industrial applications. *Processes*, Multidisciplinary Digital Publishing Institute, v. 5, n. 4, p. 56, 2017. Citado 2 vezes nas páginas 23 e 29.
- CHACHUAT, B.; MARCHETTI, A.; BONVIN, D. Process optimization via constraints adaptation. *Journal of Process Control*, Elsevier, v. 18, n. 3-4, p. 244–257, 2008. Citado na página 25.

- CHACHUAT, B.; SRINIVASAN, B.; BONVIN, D. Adaptation strategies for real-time optimization. *Computers & Chemical Engineering*, Elsevier, v. 33, n. 10, p. 1557–1567, 2009. Citado 6 vezes nas páginas 17, 21, 23, 24, 50 e 52.
- CHANONA, E. del R. et al. Modifier-adaptation schemes employing gaussian processes and trust regions for real-time optimization. *IFAC-PapersOnLine*, Elsevier, v. 52, n. 1, p. 52–57, 2019. Citado 4 vezes nas páginas 18, 27, 40 e 48.
- CHIANG, L.; LU, B.; CASTILLO, I. Big data analytics in chemical engineering. *Annual review of chemical and biomolecular engineering*, Annual Reviews, v. 8, p. 63–85, 2017. Citado na página 18.
- COSTELLO, S.; FRANÇOIS, G.; BONVIN, D. Directional real-time optimization applied to a kite-control simulation benchmark. In: IEEE. *2015 European Control Conference (ECC)*. [S.l.], 2015. p. 1594–1601. Citado na página 28.
- COSTELLO, S. et al. Modifier adaptation for run-to-run optimization of transient processes. *IFAC Proceedings Volumes*, Elsevier, v. 44, n. 1, p. 11471–11476, 2011. Citado na página 17.
- CUBILLOS, F.; ACUÑA, G.; LIMA, E. Real-time process optimization based on grey-box neural models. *Brazilian Journal of Chemical Engineering*, SciELO Brasil, v. 24, n. 3, p. 433–443, 2007. Citado na página 29.
- CUMMING, G.; FINCH, S. Inference by eye: confidence intervals and how to read pictures of data. *American psychologist*, American Psychological Association, v. 60, n. 2, p. 170, 2005. Citado na página 45.
- DARBY, M. L. et al. Rto: An overview and assessment of current practice. *Journal of Process control*, Elsevier, v. 21, n. 6, p. 874–884, 2011. Citado 4 vezes nas páginas 15, 16, 22 e 24.
- DAVIS, L. (Ed.). *Handbook of Genetic Algorithms*. [S.l.]: Van Nostrand Reinhold, 1991. Citado na página 31.
- DORMAND, J. R.; PRINCE, P. J. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, Elsevier, v. 6, n. 1, p. 19–26, 1980. Citado na página 47.
- DOWNS, J. J.; VOGEL, E. F. A plant-wide industrial process control problem. *Computers & chemical engineering*, Elsevier, v. 17, n. 3, p. 245–255, 1993. Citado na página 28.
- EFRON, B.; TIBSHIRANI, R. J. *An introduction to the bootstrap*. [S.l.]: CRC press, 1994. Citado na página 45.
- EIBEN, A. E.; SMITH, J. E. *Introduction to evolutionary computing*. [S.l.]: Springer, 2015. Citado 3 vezes nas páginas 31, 32 e 33.
- ELLIS, M.; DURAND, H.; CHRISTOFIDES, P. D. A tutorial review of economic model predictive control methods. *Journal of Process Control*, Elsevier, v. 24, n. 8, p. 1156–1178, 2014. Citado na página 16.

- FERREIRA, T. de A. et al. Real-time optimization of uncertain process systems via modifier adaptation and gaussian processes. In: IEEE. *2018 European Control Conference (ECC)*. [S.l.], 2018. p. 465–470. Citado 8 vezes nas páginas [18](#), [19](#), [27](#), [39](#), [40](#), [43](#), [47](#) e [52](#).
- FORBES, J.; MARLIN, T.; MACGREGOR, J. Model adequacy requirements for optimizing plant operations. *Computers & chemical engineering*, Elsevier, v. 18, n. 6, p. 497–510, 1994. Citado 3 vezes nas páginas [17](#), [23](#) e [25](#).
- FORBES, J. F.; MARLIN, T. E. Model accuracy for economic optimizing controllers: the bias update case. *Industrial & engineering chemistry research*, ACS Publications, v. 33, n. 8, p. 1919–1929, 1994. Citado 2 vezes nas páginas [24](#) e [25](#).
- FORBES, J. F.; MARLIN, T. E. Design cost: A systematic approach to technology selection for model-based real-time optimization systems. *Computers & chemical engineering*, Elsevier, v. 20, n. 6-7, p. 717–734, 1996. Citado na página [17](#).
- FRANCOIS, G.; BONVIN, D. Measurement-based real-time optimization of chemical processes. In: *Advances in Chemical Engineering*. [S.l.]: Elsevier, 2013. v. 43, p. 1–50. Citado na página [28](#).
- FRANÇOIS, G.; SRINIVASAN, B.; BONVIN, D. Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *Journal of Process Control*, Elsevier, v. 15, n. 6, p. 701–712, 2005. Citado na página [17](#).
- FRANÇOIS, G.; SRINIVASAN, B.; BONVIN, D. Comparison of six implicit real-time optimization schemes. *Journal européen des systèmes automatisés*, Lavoisier, v. 46, n. ARTICLE, p. 291–305, 2012. Citado na página [22](#).
- FRAZIER, P. I. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. Citado 2 vezes nas páginas [36](#) e [37](#).
- GAO, W.; WENZEL, S.; ENGELL, S. Integration of gradient adaptation and quadratic approximation in real-time optimization. In: IEEE. *2015 34th Chinese Control Conference (CCC)*. [S.l.], 2015. p. 2780–2785. Citado na página [26](#).
- GAO, W.; WENZEL, S.; ENGELL, S. Modifier adaptation with quadratic approximation in iterative optimizing control. In: IEEE. *2015 European Control Conference (ECC)*. [S.l.], 2015. p. 2527–2532. Citado na página [26](#).
- GAO, W.; WENZEL, S.; ENGELL, S. A reliable modifier-adaptation strategy for real-time optimization. *Computers & chemical engineering*, Elsevier, v. 91, p. 318–328, 2016. Citado 2 vezes nas páginas [18](#) e [26](#).
- GARDNER, J. R. et al. Bayesian optimization with inequality constraints. In: *ICML*. [S.l.: s.n.], 2014. v. 2014, p. 937–945. Citado na página [37](#).
- GELBART, M. A. *Constrained Bayesian optimization and applications*. Tese (Doutorado), 2015. Citado 7 vezes nas páginas [19](#), [35](#), [36](#), [37](#), [38](#), [41](#) e [63](#).
- GOLSHAN, M.; PISHVAIE, M. R.; BOOZARJOMEHRY, R. B. Stochastic and global real time optimization of tennessee eastman challenge problem. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 21, n. 2, p. 215–228, 2008. Citado na página [29](#).

- GOPALUNI, R. B. et al. Modern machine learning tools for monitoring and control of industrial processes: A survey. Nao publicado, 2020. Citado na página 15.
- GROS, S.; SRINIVASAN, B.; BONVIN, D. Optimizing control based on output feedback. *Computers & Chemical Engineering*, Elsevier, v. 33, n. 1, p. 191–198, 2009. Citado 2 vezes nas páginas 17 e 22.
- HAUKOOS, J. S.; LEWIS, R. J. Advanced statistics: bootstrapping confidence intervals for statistics with “difficult” distributions. *Academic emergency medicine*, Wiley Online Library, v. 12, n. 4, p. 360–365, 2005. Citado 2 vezes nas páginas 43 e 45.
- HAWKINS, D. M. The problem of overfitting. *Journal of chemical information and computer sciences*, ACS Publications, v. 44, n. 1, p. 1–12, 2004. Citado na página 35.
- HIPP, R. D. *SQLite*. 2020. Disponível em: <<https://www.sqlite.org/index.html>>. Citado na página 46.
- HUANG, C.; LI, Y.; YAO, X. A survey of automatic parameter tuning methods for metaheuristics. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 24, n. 2, p. 201–216, 2019. Citado na página 63.
- JANG, S.-S.; JOSEPH, B.; MUKAI, H. On-line optimization of constrained multivariable chemical processes. *AIChE Journal*, Wiley Online Library, v. 33, n. 1, p. 26–35, 1987. Citado na página 22.
- JONES, D. R.; SCHONLAU, M.; WELCH, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, Springer, v. 13, n. 4, p. 455–492, 1998. Citado na página 36.
- JR, E. S. G. Exponential smoothing: The state of the art. *Journal of forecasting*, Wiley Online Library, v. 4, n. 1, p. 1–28, 1985. Citado na página 39.
- KOCIJAN, J. *Modelling and control of dynamic systems using Gaussian process models*. [S.l.]: Springer, 2016. Citado na página 39.
- KRAFT, D. et al. A software package for sequential quadratic programming. DFVLR Obersfaffeuhofen, Germany, 1988. Citado 2 vezes nas páginas 32 e 53.
- KRISHNAMOORTHY, D.; FOSS, B.; SKOGESTAD, S. Real-time optimization under uncertainty applied to a gas lifted well network. *Processes*, Multidisciplinary Digital Publishing Institute, v. 4, n. 4, p. 52, 2016. Citado na página 28.
- KRSTIĆ, M.; WANG, H.-H. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, Elsevier, v. 36, n. 4, p. 595–601, 2000. Citado 2 vezes nas páginas 17 e 22.
- KRUSE, R. et al. *Computational intelligence: a methodological introduction*. [S.l.]: Springer, 2016. Citado 2 vezes nas páginas 29 e 31.
- LAGARIAS, J. C. et al. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, SIAM, v. 9, n. 1, p. 112–147, 1998. Citado na página 33.

- LASI, H. et al. Industry 4.0. *Business & information systems engineering*, Springer, v. 6, n. 4, p. 239–242, 2014. Citado na página 15.
- LU, Y. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, Elsevier, v. 6, p. 1–10, 2017. Citado na página 15.
- MARCHETTI, A.; CHACHUAT, B.; BONVIN, D. Modifier-adaptation methodology for real-time optimization. *Industrial & engineering chemistry research*, ACS Publications, v. 48, n. 13, p. 6022–6033, 2009. Citado 4 vezes nas páginas 16, 26, 27 e 39.
- MARCHETTI, A.; CHACHUAT, B.; BONVIN, D. A dual modifier-adaptation approach for real-time optimization. *Journal of Process Control*, Elsevier, v. 20, n. 9, p. 1027–1037, 2010. Citado na página 17.
- MARCHETTI, A. G. et al. Modifier adaptation as a feedback control scheme. *Industrial & Engineering Chemistry Research*, ACS Publications, 2020. Citado na página 17.
- MARCHETTI, A. G. et al. Modifier adaptation for real-time optimization—methods and applications. *Processes*, Multidisciplinary Digital Publishing Institute, v. 4, n. 4, p. 55, 2016. Citado 6 vezes nas páginas 16, 17, 25, 26, 28 e 40.
- MARLIN, T. E.; HRYMAK, A. N. et al. Real-time operations optimization of continuous processes. In: NEW YORK, NY: AMERICAN INSTITUTE OF CHEMICAL ENGINEERS, 1971-C2002. *AIChE Symposium Series*. [S.l.], 1997. v. 93, n. 316, p. 156–164. Citado na página 17.
- MILOSAVLJEVIC, P. et al. Time-optimal path-following operation in the presence of uncertainty. In: IEEE. *2016 European Control Conference (ECC)*. [S.l.], 2016. p. 2228–2233. Citado na página 28.
- MILOSAVLJEVIC, P. et al. Real-time optimization of load sharing for gas compressors in the presence of uncertainty. *Applied Energy*, Elsevier, v. 272, p. 114883, 2020. Citado na página 28.
- MOHAMMED, R. O.; CAWLEY, G. C. Over-fitting in model selection with gaussian process regression. In: SPRINGER. *International Conference on Machine Learning and Data Mining in Pattern Recognition*. [S.l.], 2017. p. 192–205. Citado na página 35.
- MONTGOMERY, D. C.; RUNGER, G. C. *Applied statistics and probability for engineers*. [S.l.]: John Wiley & Sons, 2010. Citado na página 43.
- MYUNG, I. J. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, Elsevier, v. 47, n. 1, p. 90–100, 2003. Citado na página 35.
- NARASIMHAN, S.; JORDACHE, C. *Data reconciliation and gross error detection: An intelligent use of process data*. [S.l.]: Elsevier, 1999. Citado na página 22.
- NAVIA, D. et al. Modifier-adaptation methodology for real-time optimization reformulated as a nested optimization problem. *Industrial & engineering chemistry research*, ACS Publications, v. 54, n. 48, p. 12054–12071, 2015. Citado 2 vezes nas páginas 26 e 50.
- NAVIA, D. et al. Real-time optimization for a laboratory-scale flotation column. *Computers & chemical engineering*, Elsevier, v. 86, p. 62–74, 2016. Citado na página 27.

- NAYSMITH, M.; DOUGLAS, P. Review of real time optimization in the chemical process industries. *Developments in Chemical Engineering and Mineral Processing*, Wiley Online Library, v. 3, n. 2, p. 67–87, 1995. Citado na página 17.
- NELDER, J. A.; MEAD, R. A simplex method for function minimization. *The computer journal*, Oxford University Press, v. 7, n. 4, p. 308–313, 1965. Citado na página 33.
- PANT, M. et al. Differential evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 90, p. 103479, 2020. Citado 2 vezes nas páginas 32 e 33.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, JMLR. org, v. 12, p. 2825–2830, 2011. Citado na página 51.
- PETSAGKOURAKIS, P.; CHACHUAT, B.; RIO-CHANONA, E. A. del. Safe real-time optimization using multi-fidelity gaussian processes. *arXiv preprint arXiv:2111.05589*, 2021. Citado na página 28.
- PETSAGKOURAKIS, P. et al. Constrained reinforcement learning for dynamic optimization under uncertainty. *arXiv preprint arXiv:2006.02750*, 2020. Citado na página 18.
- QUELHAS, A. D.; JESUS, N. J. C. de; PINTO, J. C. Common vulnerabilities of rto implementations in real chemical processes. *The Canadian Journal of Chemical Engineering*, Wiley Online Library, v. 91, n. 4, p. 652–668, 2013. Citado 4 vezes nas páginas 18, 29, 42 e 63.
- RAO, S. S. *Engineering optimization: theory and practice*. [S.l.]: John Wiley & Sons, 2019. Citado na página 31.
- RICKER, N. L. Decentralized control of the tennessee eastman challenge process. *Journal of process control*, Elsevier, v. 6, n. 4, p. 205–221, 1996. Citado na página 29.
- RIO-CHANONA, E. A. del et al. Modifier adaptation meets bayesian optimization and derivative-free optimization. *arXiv preprint arXiv:2009.08819*, 2020. Citado 8 vezes nas páginas 18, 19, 28, 29, 41, 42, 48 e 53.
- ROBERTS, P. An algorithm for steady-state system optimization and parameter estimation. *International Journal of Systems Science*, Taylor & Francis, v. 10, n. 7, p. 719–734, 1979. Citado na página 24.
- ROBERTS, P. Coping with model-reality differences in industrial process optimisation—a review of integrated system optimisation and parameter estimation (isope). *Computers in Industry*, Elsevier, v. 26, n. 3, p. 281–290, 1995. Citado na página 25.
- RUPPEN, D.; BONVIN, D.; RIPPIN, D. Implementation of adaptive optimal operation for a semi-batch reaction system. *Computers & chemical engineering*, Elsevier, v. 22, n. 1-2, p. 185–199, 1998. Citado 2 vezes nas páginas 28 e 49.
- SAVAGE, T. et al. Data-driven optimization for process systems engineering applications. *Chemical Engineering Science*, Elsevier, p. 117135, 2021. Citado na página 22.

- SCHWAAB, M. et al. Nonlinear parameter estimation through particle swarm optimization. *Chemical Engineering Science*, Elsevier, v. 63, n. 6, p. 1542–1552, 2008. Citado na página 23.
- SERRALUNGA, F. J.; MUSSATI, M. C.; AGUIRRE, P. A. Model adaptation for real-time optimization in energy systems. *Industrial & Engineering Chemistry Research*, ACS Publications, v. 52, n. 47, p. 16795–16810, 2013. Citado na página 28.
- SHUKLA, H. A. et al. Convergence certificate for stochastic derivative-free trust-region methods based on gaussian processes. *arXiv preprint arXiv:2010.01120*, 2020. Citado 2 vezes nas páginas 28 e 41.
- SINGER, S.; NELDER, J. Nelder-mead algorithm. *Scholarpedia*, v. 4, n. 7, p. 2928, 2009. Citado 2 vezes nas páginas 33 e 34.
- SINGHAL, M.; FAULWASSER, T.; BONVIN, D. On handling cost gradient uncertainty in real-time optimization. *IFAC-PapersOnLine*, Elsevier, v. 48, n. 8, p. 176–181, 2015. Citado na página 26.
- SINGHAL, M. et al. Real-time optimization based on adaptation of surrogate models. *IFAC-PapersOnLine*, Elsevier, v. 49, n. 7, p. 412–417, 2016. Citado 2 vezes nas páginas 26 e 47.
- SKOGESTAD, S. Self-optimizing control: The missing link between steady-state optimization and control. *Computers & Chemical Engineering*, Elsevier, v. 24, n. 2-7, p. 569–575, 2000. Citado 2 vezes nas páginas 17 e 22.
- SRINIVASAN, B.; BONVIN, D. Real-time optimization of batch processes by tracking the necessary conditions of optimality. *Industrial & engineering chemistry research*, ACS Publications, v. 46, n. 2, p. 492–504, 2007. Citado na página 22.
- SRINIVASAN, B.; BONVIN, D. 110th anniversary: A feature-based analysis of static real-time optimization schemes. *Industrial & Engineering Chemistry Research*, ACS Publications, v. 58, n. 31, p. 14227–14238, 2019. Citado 2 vezes nas páginas 17 e 47.
- TATJEWSKI, P. Iterative optimizing set-point control—the basic principle redesigned. *IFAC Proceedings Volumes*, Elsevier, v. 35, n. 1, p. 49–54, 2002. Citado 2 vezes nas páginas 25 e 26.
- VIRTANEN, P. et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, Nature Publishing Group, v. 17, n. 3, p. 261–272, 2020. Citado 2 vezes nas páginas 47 e 51.
- WEISSMAN, S. A.; ANDERSON, N. G. Design of experiments (doe) and process optimization. a review of recent publications. *Organic Process Research & Development*, ACS Publications, v. 19, n. 11, p. 1605–1633, 2015. Citado na página 23.
- WILLIAMS, C. K.; RASMUSSEN, C. E. *Gaussian processes for machine learning*. [S.l.]: MIT press Cambridge, MA, 2006. v. 2. Citado 3 vezes nas páginas 27, 34 e 35.
- WILLIAMS, T. J.; OTTO, R. E. A generalized chemical processing model for the investigation of computer control. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, IEEE, v. 79, n. 5, p. 458–473, 1960. Citado 2 vezes nas páginas 28 e 47.



- YIP, W. S.; MARLIN, T. E. Multiple data sets for model updating in real-time operations optimization. *Computers & chemical engineering*, Elsevier, v. 26, n. 10, p. 1345–1362, 2002. Citado na página 17.
- ZENDEHBOUDI, S.; REZAEI, N.; LOHI, A. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. *Applied energy*, Elsevier, v. 228, p. 2539–2566, 2018. Citado na página 18.
- ZHANG, Y.; FORBES, J. F. Extended design cost: a performance criterion for real-time optimization systems. *Computers & Chemical Engineering*, Elsevier, v. 24, n. 8, p. 1829–1841, 2000. Citado 2 vezes nas páginas 47 e 48.
- ZHANG, Y.; MONDER, D.; FORBES, J. F. Real-time optimization under parametric uncertainty: a probability constrained approach. *Journal of Process control*, Elsevier, v. 12, n. 3, p. 373–389, 2002. Citado na página 15.
- ZHANG, Z. et al. Real-time optimization and control of nonlinear processes using machine learning. *Mathematics*, Multidisciplinary Digital Publishing Institute, v. 7, n. 10, p. 890, 2019. Citado na página 18.