



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA - CCT  
DEPARTAMENTO DE ENGENHARIA QUÍMICA – DEQ  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA - PPEQ**

**TRABALHO DE TESE**

**ALGORITMO HEURÍSTICO DE RETROALIMENTAÇÃO INCLUSIVA PARA  
REGRESSÃO DE DADOS DE PROCESSO**

Thalita Cristine Ribeiro Lucas Fernandes

Orientador: Prof. Dr. Antonio Carlos Brandão de Araujo

**CAMPINA GRANDE - PARAÍBA**

**2022**

**THALITA CRISTINE RIBEIRO LUCAS FERNANDES**

**ALGORITMO HEURÍSTICO DE RETROALIMENTAÇÃO INCLUSIVA PARA  
REGRESSÃO DE DADOS DE PROCESSO**

Trabalho de Tese apresentado ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal de Campina Grande, como parte dos requisitos necessários para obtenção do título de Doutora em Engenharia Química.

**Orientador:** Prof. Dr. Antonio Carlos Brandão de Araujo


**CAMPINA GRANDE - PARAÍBA**

**2022**

|       |  |
|-------|--|
| F363a | <p>Fernandes, Thalita Cristine Ribeiro Lucas.</p> <p>Algoritmo heurístico de retroalimentação inclusiva para regressão de dados de processo / Thalita Cristine Ribeiro Lucas Fernandes. – Campina Grande, 2022.</p> <p>190 f. : il. color.</p> <p>Tese (Doutorado em Engenharia Química) – Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia, 2022.</p> <p>"Orientação: Prof. Dr. Antonio Carlos Brandão de Araujo".</p> <p>Referências.</p> <p>1. Processos Químicos. 2. Modelos Substitutos. 3. Modelagem e Simulação de Processos. 4. Algoritmo. 4. Automação. 5. Aprendizado de Máquina. I. Araujo, Antonio Carlos Brandão de. II. Título.</p> <p style="text-align: right;">CDU 66.011(043)</p> |
|-------|--|

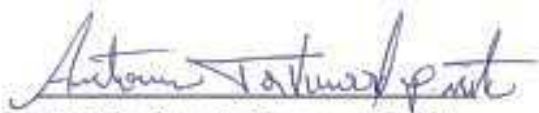
**ALGORITMO HEURÍSTICO DE RETROALIMENTAÇÃO INCLUSIVA PARA  
REGRESSÃO DE DADOS DE PROCESSO**

**BANCA EXAMINADORA**



---

Prof. Dr. Antonio Carlos Brandão de Araujo  
Orientador (DEQ/CCT/UFCG)



---

Prof. Dr. Antonio Tavernard P. Neto  
Examinador Interno  
(DEQ/CCT/UFCG)



---

Prof. Dr. Sidinei Kleber da Silva  
Examinador Externo  
(DEQ/CCT/UFCG)



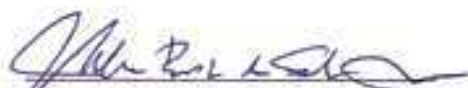
---

Prof. Dr. José Jailson Nicácio Alves  
Examinador Interno  
(DEQ/CCT/UFCG)



---

Prof. Dr. Vimário Simões Silva  
Examinador Externo  
(DEQ/CCT/UFCG)



---

Prof. Dr. Heleno Bispo da S. Júnior  
Examinador Interno  
(DEQ/CCT/UFCG)

**CAMPINA GRANDE - PARAÍBA**

**Data da Defesa: 27/04/2022**

## **AGRADECIMENTOS**

Ao Professor Antonio Carlos Brandão de Araújo pelo conhecimento transmitido e pela orientação.

À CAPES pelo financiamento da bolsa durante o período de realização deste trabalho.

A todas as demais pessoas que contribuíram positivamente de forma direta ou indireta para a conclusão deste trabalho.

Resumo do Trabalho de Tese apresentado ao DEQ/UFMG como parte dos requisitos necessários para a obtenção do título de Doutora em Engenharia Química.

Thalita Cristine Ribeiro Lucas Fernandes

Abril/2022

Este trabalho apresenta um algoritmo de aprendizado de máquina automatizado (AutoML) sistemático e simples. A principal contribuição é produzir o modelo de regressão mais simples, sempre que possível (ex.: modelo de regressão polinomial de segunda ordem via seleção de recursos sequenciais baseado nos mínimos quadrados) ou então, gerar modelos não lineares mais complexos (ex.: regressão gaussiana). O algoritmo é capaz de produzir estes resultados usando técnicas de design sequencial para preencher habilmente o espaço amostral com pontos “interessantes”, gerando um conjunto de dados que é utilizado para selecionar o modelo de regressão mais simples possível. Esse modelo mais simples é gerado de forma iterativa a partir de um conjunto predefinido de modelos de regressão candidatos. O objetivo é minimizar o número de chamadas para o processo gerador (simulador), resultando no menor número de amostras. Cada conjunto de dados produzidos iterativamente é usado de forma exaustiva e eficaz, capaz de convergir até mesmo respostas difíceis que requerem um grande número de amostras. A aplicação do algoritmo proposto em casos importantes (equações matemáticas de difícil resolução, coluna de destilação em Aspen Plus e uma Planta de tratamento de efluentes em Simulink) mostra sua efetividade na construção de metamodelos com capacidade preditiva significativa. É sugerida a utilização de técnicas de regressão puramente não lineares em situações que as simulações demandem mais tempo do que o processamento do algoritmo. Em geral, um mix de métodos de regressão linear e não linear para a construção dos metamodelos é recomendada para a maioria dos casos, para compensar o tempo de processamento e a capacidade preditiva.

**Palavras-chave:** Modelos substitutos; Algoritmo; Aprendizado de Máquina; Automação; Processos químicos.

Abstract of the Thesis presented to the DEQ / UFCG as part of the requirements for obtaining a Doctor's degree in Chemical Engineering.

Thalita Cristine Ribeiro Lucas Fernandes

April/2022

This is an attempt to create a simple, but quite systematic, automated machine learning (AutoML) algorithm. The main contribution is to produce the simplest regression model (e.g., second order polynomial regression model via OLS based sequential feature selection) whenever possible, or else generate more complex, and therefore less desirable, nonlinear (e.g., gaussian process regression) models. It does so by efficiently using sequential design techniques to cleverly fill the sample space with “interesting” points, generating a dataset (which includes the responses obtained by “querying” the actual underlying process) on demand that is used to select the simplest possible regression model, among a predefined set of candidate regression models, in an iteratively way until particular convergence criteria are met. The intended goal is therefore to minimize the number of calls to the generating process, resulting in the least number of samples. Each dataset produced iteratively is exhaustively and effectively used up in an effort to converge even difficult responses that have not met the criteria even with a large number of samples. Application of the proposed algorithm to important cases shows its effectiveness in building metamodels with significant predictive capabilities. It is suggested the use of pure nonlinear regression techniques in situations in which data takes more time to gather than to be processed by the algorithm. In general, a carefully chosen mix of both linear and nonlinear regression methods to metamodel building is recommended for most cases, as a tradeoff between processing time and predictive capacity.

**Keywords:** Substitute models; Algorithm; Machine Learning; Automation; Chemical processes.

## SUMÁRIO

|           |  |           |
|-----------|--|-----------|
| <b>1.</b> | <b>INTRODUÇÃO .....</b>  | <b>1</b>  |
| <b>2.</b> | <b>FUNDAMENTAÇÃO TEÓRICA.....</b>                                    | <b>3</b>  |
| 2.1.      | <b>Aprendizado de máquina automatizado.....</b>                      | <b>3</b>  |
| 2.1.1.    | <b>Técnicas de abordagens do aprendizado de máquina .....</b>        | <b>3</b>  |
| 2.1.2.    | <b>Softwares com algoritmos de aprendizado de máquina .....</b>      | <b>6</b>  |
| 2.1.2.1.  | <b>Auto Weka.....</b>  | <b>6</b>  |
| 2.1.2.2.  | <b>AutoGluon – Tabular.....</b>                                      | <b>7</b>  |
| 2.1.2.3.  | <b>H2O AutoML .....</b>  | <b>8</b>  |
| 2.1.2.4.  | <b>Sumo Toolbox.....</b>   | <b>9</b>  |
| 2.1.2.5.  | <b>Metodologia ALAMO.....</b>  | <b>10</b> |
| 2.2.      | <b>Técnicas de Amostragem.....</b>                                   | <b>11</b> |
| 2.2.1.    | <b>Técnicas de Amostragem para tamanho de amostras fixo .....</b>    | <b>12</b> |
| 2.2.1.1.  | <b>Análise Fatorial .....</b>  | <b>12</b> |
| 2.2.1.2.  | <b>O conceito de discrepância.....</b>                               | <b>15</b> |
| 2.2.1.3.  | <b>Sequências de Sobol .....</b>                                     | <b>15</b> |
| 2.2.1.4.  | <b>Sequências de Halton.....</b>                                     | <b>17</b> |
| 2.2.1.5.  | <b>Sequências de Hammersley .....</b>                                | <b>18</b> |
| 2.2.1.6.  | <b>Voronoi .....</b>   | <b>18</b> |
| 2.2.1.7.  | <b>Niederreiter .....</b>  | <b>19</b> |
| 2.2.1.8.  | <b>Sequências de Faure .....</b>                                     | <b>20</b> |
| 2.2.1.9.  | <b>Latin Hipercubo .....</b>   | <b>21</b> |
| 2.2.1.10. | <b>Máxima Entropia.....</b>  | <b>22</b> |
| 2.2.1.11. | <b>Minimax e Maximin.....</b>  | <b>22</b> |
| 2.2.1.12. | <b>Amostragem Experimental Uniforme (UD) .....</b>                   | <b>23</b> |
| 2.2.2.    | <b>Técnicas de Amostragem Sequencial.....</b>                        | <b>24</b> |
| 2.2.2.1.  | <b>Lola-Voronoi.....</b>   | <b>26</b> |
| 2.2.2.2.  | <b>SED Toolbox.....</b>  | <b>27</b> |
| 2.2.2.3.  | <b>Intersite-projected .....</b>                                     | <b>27</b> |
| 2.2.2.4.  | <b>Intersite-projected-threshold .....</b>                           | <b>27</b> |
| 2.2.2.5.  | <b>Optimizer-projected .....</b>                                     | <b>28</b> |
| 2.2.2.6.  | <b>Optimizer-intersite.....</b>                                      | <b>29</b> |
| 2.3.      | <b>Transformações de variáveis dependentes e independentes .....</b> | <b>30</b> |
| 2.4.      | <b>Otimização Substituta.....</b>                                    | <b>32</b> |
| 2.5.      | <b>Modelos Substitutos .....</b>                                     | <b>33</b> |



|         |  |    |
|---------|--|----|
| 2.5.1.  | Funções de Base Radial .....   | 34 |
| 2.5.2.  | Modelos de Regressão Linear Múltipla.....  | 35 |
| 2.5.3.  | Kriging.....   | 36 |
| 2.6.    | Construção de modelos substitutos disponíveis no MATLAB® .....                       | 42 |
| 2.7.    | Medidas de um bom modelo .....   | 44 |
| 2.7.1.  | Correlação máxima de pares .....   | 44 |
| 2.7.2.  | Discrepância L2 modificada.....  | 46 |
| 2.8.    | Detecção do estado estacionário .....  | 47 |
| 2.9.    | Caixa de ferramentas GPML .....  | 49 |
| 2.10.   | Validação Cruzada .....  | 51 |
| 2.10.1. | K-fold .....   | 51 |
| 2.10.2. | Holdout.....   | 52 |
| 2.10.3. | Validação de subamostragem aleatória repetida.....                                   | 52 |
| 2.10.4. | Leave-p-out.....   | 52 |
| 2.10.5. | Leave-one-out.....   | 52 |
| 2.11.   | Método dos mínimos quadrados (OLS) com seleção de variáveis .....                    | 53 |
| 2.12.   | Otimização Bayesiana.....  | 54 |
| 2.13.   | Técnicas de interpolação .....   | 56 |
| 2.13.1. | Método de interpolação Akima .....   | 57 |
| 3.      | ALGUNS TRABALHOS DESENVOLVIDOS .....   | 59 |
| 3.1.    | Otimização Substituta.....   | 59 |
| 3.2.    | Modelos Substitutos .....  | 60 |
| 3.3.    | Técnicas de transformações de variáveis .....  | 63 |
| 3.4.    | Validação Cruzada.....   | 63 |
| 3.5.    | Técnicas de Amostragem.....  | 64 |
| 3.6.    | Aprendizado de Máquina.....  | 67 |
| 3.7.    | Outros estudos na área .....   | 68 |
| 4.      | METODOLOGIA .....  | 69 |
| 4.1.    | Comunicação do simulador gerador de dados com o algoritmo .....                      | 75 |
| 4.2.    | Definição da estrutura do problema e seleção do método de amostragem sequencial..... | 76 |
| 4.3.    | Definição de alguns parâmetros .....   | 77 |
| 4.4.    | Cálculo de alguns parâmetros internos .....  | 79 |
| 4.5.    | Manipulação das variáveis de entrada e saída .....                                   | 80 |
| 4.6.    | Inicialização de matrizes .....  | 84 |

|           |   |     |
|-----------|---|-----|
| 4.7.      | Geração de pontos iniciais (m) .....  | 84  |
| 4.8.      | Cálculo do valor inicial do “Erro” para todas as respostas ( $Erro_m$ ) ..... | 85  |
| 4.9.      | Cálculo valor do $Erro_r$ .....   | 85  |
| 4.10.     | Escolha da resposta que será processada (kresp) .....                         | 85  |
| 4.11.     | Cálculo dos parâmetros do controlador .....                                   | 86  |
| 4.12.     | Cálculo do número de amostras que serão acrescentadas .....                   | 86  |
| 4.13.     | Construção dos metamodelos para as respostas .....                            | 87  |
| 4.14.     | Crítérios de saída das variáveis .....  | 88  |
| 4.15.     | Funções utilizadas no algoritmo .....   | 90  |
| 4.15.1.   | VarTransModel .....   | 90  |
| 4.15.2.   | “Predictors” .....  | 94  |
| 4.15.3.   | fscoreGpml .....  | 96  |
| 4.15.4.   | varTransModelOptimization .....   | 97  |
| 4.15.5.   | varTransObj .....   | 100 |
| 4.15.6.   | funVarTrans .....   | 101 |
| 4.15.7.   | simulFuncAspenPlus .....  | 102 |
| 4.15.8.   | simulFuncSimulink .....   | 103 |
| 4.15.9.   | simulFuncMatlab .....   | 104 |
| 4.15.9.1. | Função de Rastrigin .....   | 104 |
| 4.15.9.2. | Função de Schwefel .....  | 105 |
| 4.15.9.3. | Função hiperelipsoide rotacionada .....                                       | 106 |
| 4.15.9.4. | Função de Styblinski-Tang .....   | 106 |
| 4.15.9.5. | Função de Zharakov .....  | 107 |
| 4.15.9.6. | Função da soma dos quadrados .....  | 107 |
| 4.15.9.7. | Função de Ackley .....  | 108 |
| 4.15.9.8. | Função autoral .....  | 108 |
| 4.15.10.  | standNorm .....   | 109 |
| 4.15.11.  | checkpoint .....  | 110 |
| 4.15.12.  | myPredict .....   | 111 |
| 4.15.13.  | uniqueTolComumns .....  | 113 |
| 4.15.14.  | licols .....  | 115 |
| 5.        | RESULTADOS E DISCUSSÃO .....  | 116 |
| 5.1.      | Aplicação da metodologia em equações matemáticas diretamente no MATLAB .....  | 118 |
| 5.2.      | Aplicação da metodologia em uma planta de destilação em Aspen Plus .....      |     |

|  |            |
|--|------------|
| <b>5.3. Aplicação da metodologia a uma planta de tratamento de efluentes em Simulink .....</b> | <b>120</b> |
| <b>5.4. Resultados .....</b>   | <b>122</b> |
| <b>6. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS .....</b>                                  | <b>168</b> |
| <b>7. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>   | <b>170</b> |
| <b>APÊNDICE A. ....</b>  | <b>180</b> |
| <b>ANEXO I .....</b>   | <b>186</b> |

## LISTA DE FIGURAS

|   |     |
|---|-----|
| <b>Figura 1: Sequência de cálculo do algoritmo</b> Fonte: Bhattacharyya (2008).....   | 29  |
| <b>Figura 2: Dados de um processo, com ruído e desvios.</b> Fonte: Rhinehart (2013) 48  |     |
| <b>Figura 3: Função de Rastrigin.</b> Fonte: Surjanovic e Bingham (2013).....   | 105 |
| <b>Figura 4: Função de Schwefel.</b> Fonte: Surjanovic e Bingham (2013).....  | 105 |
| <b>Figura 5: Função hiperelipsoide rotacionada.</b> Fonte: Surjanovic e Bingham (2013).....   | 106 |
| <b>Figura 6: Função de Styblinski-Tang.</b> Fonte: Surjanovic e Bingham (2013). .....   | 106 |
| <b>Figura 7: Função de Zhakarov.</b> Fonte: Surjanovic e Bingham (2013).....  | 107 |
| <b>Figura 8: Função da soma dos quadrados.</b> Fonte: Surjanovic e Bingham (2013). .....  | 107 |
| <b>Figura 9: Função de Ackley.</b> Fonte: Surjanovic e Bingham (2013). .....  | 108 |
| <b>Figura 10: Função Autoral</b> Fonte: Próprio Autor .....   | 109 |
| <b>Figura 11: Coluna de destilação no Aspen Plus</b> Fonte: Próprio Autor. Simulação no Aspen.....  | 119 |
| <b>Figura 12: Esquema da planta de tratamento de efluentes</b> Fonte: Jeppson et al. (2011).....  | 121 |
| <b>Figura 13: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no MATLAB (Caso 1).</b> Fonte: Próprio Autor. ....     | 123 |
| <b>Figura 14: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no MATLAB (Caso 2).</b> Fonte: Próprio Autor. ....     | 125 |
| <b>Figura 15: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no MATLAB (Caso 3).</b> Fonte: Próprio Autor. ....     | 127 |
| <b>Figura 16: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no MATLAB (Caso 4).</b> Fonte: Próprio Autor. ....     | 129 |
| <b>Figura 17: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Aspen Plus (Caso 1).</b> Fonte: Próprio Autor. .... | 131 |
| <b>Figura 18: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Aspen Plus (Caso 2).</b> Fonte: Próprio Autor ..... | 133 |
| <b>Figura 19: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Aspen Plus (Caso 3).</b> Fonte: Próprio Autor. .... | 135 |
| <b>Figura 20: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Aspen Plus (Caso 4).</b> Fonte: Próprio Autor. .... | 137 |
| <b>Figura 21: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Simulink (Caso 1).</b> Fonte: Próprio Autor. ....   | 139 |
| <b>Figura 22: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Simulink (Caso 2).</b> Fonte: Próprio Autor. ....   | 141 |
| <b>Figura 23: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Simulink (Caso 3).</b> Fonte: Próprio Autor. ....   | 143 |
| <b>Figura 24: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Simulink (Caso 4).</b> Fonte: Próprio Autor. ....   | 145 |
| <b>Figura 25: Evolução do processo iterativo do algoritmo aplicado às equações matemáticas em Matlab (Caso 1).</b> Fonte: Próprio Autor .....           | 147 |
| <b>Figura 26: Evolução do processo iterativo do algoritmo aplicado às equações matemáticas em Matlab (Caso 2).</b> Fonte: Próprio Autor .....           | 148 |
| <b>Figura 27: Evolução do processo iterativo do algoritmo aplicado às equações matemáticas em Matlab (Caso 3).</b> Fonte: Próprio Autor .....           | 149 |

|  |     |
|--|-----|
| <b>Figura 28: Evolução do processo iterativo do algoritmo aplicado às equações matemáticas em Matlab (Caso 4). Fonte: Próprio Autor</b> .....                                | 150 |
| <b>Figura 29: Evolução do processo iterativo do algoritmo aplicado à simulação de uma coluna de destilação em Aspen Plus (Caso 1). Fonte: Próprio Autor</b> .....            | 151 |
| <b>Figura 30: Evolução do processo iterativo do algoritmo aplicado à simulação de uma coluna de destilação em Aspen Plus (Caso 2). Fonte: Próprio Autor</b> .....            | 152 |
| <b>Figura 31: Evolução do processo iterativo do algoritmo aplicado à simulação de uma coluna de destilação em Aspen Plus (Caso 3). Fonte: Próprio Autor</b> .....            | 153 |
| <b>Figura 32: Evolução do processo iterativo do algoritmo aplicado à simulação de uma coluna de destilação em Aspen Plus (Caso 4). Fonte: Próprio Autor</b> .....            | 154 |
| <b>Figura 33: Evolução do processo iterativo do algoritmo aplicado à simulação de uma planta de tratamento de efluentes em Simulink (Caso 1). Fonte: Próprio Autor</b> ..... | 155 |
| <b>Figura 34: Evolução do processo iterativo do algoritmo aplicado à simulação de uma planta de tratamento de efluentes em Simulink (Caso 2). Fonte: Próprio Autor</b> ..... | 156 |
| <b>Figura 35: Evolução do processo iterativo do algoritmo aplicado à simulação de uma planta de tratamento de efluentes em Simulink (Caso 3). Fonte: Próprio Autor</b> ..... | 157 |
| <b>Figura 36: Evolução do processo iterativo do algoritmo aplicado à simulação de uma planta de tratamento de efluentes em Simulink (Caso 4). Fonte: Próprio Autor</b> ..... | 158 |
| <b>Figura 37: Tempo de processamento do algoritmo para todos os casos estudados. Fonte: Próprio Autor</b> .....  | 163 |
| <b>Figura 38: Porcentagem de respostas convergidas para todos os casos estudados</b> .....   | 164 |
| <b>Figura 39: Porcentagem de respostas com o valor do <math>Q^2</math> acima do limite inferior de 0,97 para todos os casos. Fonte: Próprio Autor</b> .....                  | 164 |
| <b>Figura 40: Número total de amostras necessárias para a construção dos metamodelos em cada um dos casos estudados. Fonte: Próprio Autor</b> .....                          | 165 |
| <b>Figura 41: Número mínimo de amostras necessárias para a construção dos metamodelos em cada um dos casos estudados. Fonte: Próprio Autor</b> .....                         | 166 |

## LISTA DE TABELAS

|   |            |
|---|------------|
| <b>Tabela 1: Especificações da corrente F1 Fonte: Próprio Autor .....</b>                             | <b>119</b> |
| <b>Tabela 2: Especificações da corrente F2 Fonte: Próprio autor .....</b>                             | <b>119</b> |
| <b>Tabela 3: Iterações para o exemplo do Matlab (Caso 2). Fonte: Próprio Autor ...</b>                | <b>160</b> |
| <b>Tabela 4: Respostas processadas no algoritmo de otimização “kresp”. Fonte: Próprio Autor .....</b> | <b>167</b> |
| <b>Tabela 5: Matlab Caso1 .....</b>   | <b>180</b> |
| <b>Tabela 6: Matlab Caso 2 (Lola-Voronoi).....</b>  | <b>180</b> |
| <b>Tabela 7: Matlab Caso 3 (100% Linear).....</b>   | <b>181</b> |
| <b>Tabela 8: Matlab Caso 4 (100% não linear) .....</b>  | <b>181</b> |
| <b>Tabela 9: Aspen Plus Caso 1 .....</b>  | <b>182</b> |
| <b>Tabela 10: Aspen Plus Caso 2 (Lola-Voronoi).....</b>   | <b>182</b> |
| <b>Tabela 11: Aspen Plus Caso 3 (100% Linear) .....</b>   | <b>183</b> |
| <b>Tabela 12: Aspen Plus Caso 4 (100% Não linear) .....</b>   | <b>183</b> |
| <b>Tabela 13: Simulink Caso 1 .....</b>   | <b>184</b> |
| <b>Tabela 14: Simulink Caso 2 (Lola-Voronoi).....</b>   | <b>184</b> |
| <b>Tabela 15: Simulink Caso 3 (100% linear) .....</b>   | <b>185</b> |
| <b>Tabela 16: Simulink Caso 4 (100% não linear) .....</b>   | <b>185</b> |

## 1. INTRODUÇÃO

Muitos processos de engenharia necessitam de modelos computacionais para que seja possível a realização de alguns estudos. Existem vários softwares capazes de processar grandes simulações, permitindo que o usuário possa testar e tomar decisões de modificações para uma possível melhoria, por exemplo, em uma planta da indústria química.

Dependendo do tipo de processo a ser estudado e da robustez da modelagem do mesmo, demanda-se um esforço computacional intenso, o que acarreta também em simulações bastante demoradas. Quanto mais rigor tiver a modelagem, maior será o tempo de processamento da mesma, conseqüentemente, se for necessária a realização de uma análise de sensibilidade, o usuário poderá passar longas horas ou até mesmo dias, a espera de resultados.

Diante desta demanda de cada vez existirem processos mais complexos e da necessidade de solucionar problemas em menos tempo, é interessante que se utilizem técnicas que sejam capazes de construir modelos que representem tais processos e possam ser utilizados para testar novas situações, e assim, ajudar o usuário a tomar decisões em menos tempo. Estes modelos podem receber o nome de metamodelo ou modelos substitutos.

Existem dois termos utilizados para encontrar um modelo que descreva a relação entre as variáveis independentes e o resultado: Inferência e predição. A inferência utiliza o modelo para aprender sobre o processo de geração de dados e a previsão para prever os resultados para novos pontos de dados (WANG, McCORMICK e LEEK, 2020).

A diferença principal entre os modelos adequados para inferência está na interpretabilidade. Apenas alguns métodos interpretáveis são úteis para inferência. A escolha entre inferência e predição vai depender do objetivo final do usuário. Broelman (2001) apresenta o procedimento de cálculo de cada um desses dois termos, conforme apresentado a seguir.

Em um procedimento de inferência, o primeiro passo é a realização da modelagem. Na etapa de modelagem deve-se justificar o processo de geração

dos dados e escolher o modelo estocástico que melhor se aproxima. A segunda etapa consiste na validação do modelo onde deve-se avaliá-lo utilizando a análise residual ou testes de qualidade de ajuste e, por último, realizar a etapa de inferência, na qual utiliza-se o modelo estocástico escolhido para entender o processo de geração de dados.

Na predição, a primeira etapa também é chamada de modelagem, entretanto, deve-se considerar vários modelos e configurações de parâmetros diferentes. A segunda etapa consiste na seleção do modelo onde deve-se identificar o modelo com maior desempenho preditivo por meio da utilização de conjuntos de validação e teste. Por último, na etapa de predição, deve-se aplicar o modelo selecionado em novos dados com a expectativa de que o modelo seja capaz de prever resultados em dados não vistos.

Encontrar um modelo substituto adequado, ou metamodelo, não é uma tarefa fácil pois demanda muito conhecimento teórico e de programação. E, considerando a realidade de uma planta química, nem todos os funcionários possuem tais conhecimentos.

Em virtude disto, este trabalho tem o objetivo geral de desenvolver um procedimento sistemático de cunho heurístico para obtenção de metamodelos preditivos a partir da regressão de dados determinísticos oriundos de simulações de plantas de processos. Como objetivos secundários tem-se:

- Definir o planejamento experimental;
- Definir a estrutura dos metamodelos;
- Definir o Algoritmo para obtenção dos modelos preditivos;
- Aplicar o procedimento de coleta de dados;
- Testar os metamodelos com novos dados.



## **2. FUNDAMENTAÇÃO TEÓRICA**

### **2.1. Aprendizado de máquina automatizado**

O aprendizado de máquina é o processo no qual problemas do mundo real são tomados como reerência para o desenvolvimento de rotinas automatizadas. Esta técnica inclui todos os estágios, desde a criação de um conjunto de dados brutos até a criação de um modelo pronto para implementação (THOMTON et al., 2013). A aplicação desta ferramenta facilita a utilização por parte de usuários que não tenham intimidade com o processo de criação.

As etapas que incluem o desenvolvimento de um aprendizado de máquina automatizado podem ser: Preparação de dados, seleção de recursos, detecção e tratamento de dados, seleção do modelo, otimização de hiperparâmetros, análise dos resultados obtidos e validação.

A aplicação do aprendizado de máquinas é muito ampla. Pode ser utilizado para filtragem de spam, motores de busca, bioinformática, reconhecimento de fala e escrita, entre outros (WERNICK et al., 2010). Em análise de dados, este método permite que usuários sejam capazes de produzir decisões e resultados confiáveis através do aprendizado das relações e tendências históricas nos dados.

#### **2.1.1. Técnicas de abordagens do aprendizado de máquina**

Neste tópico trataremos sobre algumas técnicas de abordagens do aprendizado de máquina de forma bem geral. A Primeira se chama aprendizado por regras de associação e trata-se de um método para encontrar padrões entre conjuntos de dados quando se tem uma enorme quantidade desses conjuntos (MENZIES e HU, 2003). A aplicação mais comum deste tipo de abordagem é em análise de transações de compras. Algumas métricas podem ser utilizadas e as restrições mais conhecidas na literatura e utilizadas são suporte e confiança (HIPPI, GUNTZER e NAKHAEIZADEH, 2000).

Dado um conjunto que contenha A e B, a sua medida suporte corresponde à porcentagem de transações de compras da base de dados que contém os itens

A e B, mostrando uma relevância. A confiança representa, dentre as transações que possuem os itens de A, a porcentagem de transações que também apresenta os itens de B, sendo assim uma validação da regra (Regras de Associação, 2019).

A próxima técnica é conhecida como rede neural artificial e trata-se de um algoritmo que possui um sistema de “neurônios” interconectados, simulando o comportamento de redes neurais biológicas (HARDICK, 2021). Algumas principais aplicações são: Sensoriamento remoto, diagnóstico médico, processamento de voz, biometria, análise de dados, entre outros.

Conforme mostrado por Hertz, Palmer e Krogh (1990), o algoritmo funciona a partir da modificação dos pesos das conexões entre os neurônios onde os pesos iniciais são modificados de forma iterativa seguindo algum critério mostrado a seguir:

- **Supervisionado:** É apresentado um conjunto de treino que contém as entradas e as saídas desejadas;
- **Reforço:** Em cada entrada, é produzida uma indicação sobre a adequação das saídas correspondentes;
- **Não-Supervisionado:** A rede atualiza sem a utilização de pares de entrada e saída. Consequentemente, também não conta com a indicação das saídas produzidas.

Uma outra técnica é chamada de aprendizado profundo e se caracteriza pela parte mais abrangente baseada na aprendizagem de representações de dados, onde de acordo com Deng e Yu (2014), utiliza uma cascata de camadas diferentes de unidades de processamento não linear para a extração e transformação de características e aprendem vários níveis de representações que correspondem a diferentes níveis hierárquicos. Esse tipo de abordagem se baseia na ideia de fatores explicativos hierárquicos, onde os conceitos de nível superior são aprendidos a partir do nível mais baixo (BENGIO, COURVILLE e VINCENT, 2013).

Existem dois tipos de interpretações, uma que é baseada no teorema da aproximação universal, onde a capacidade de rede neural de alimentação é

direta com uma única camada de tamanho finito e aproximada por funções contínuas (CYBENKO, 1989). E a outra é baseada na probabilidade, a qual inclui inferência, conceitos de otimização como treinamento e testes, relacionados à adaptação e generalização (DENG e YU, 2014).

Ainda podemos abordar a lógica de programação indutiva, que faz uso de programação lógica com uma representação uniforme, para exemplos de *inputs*, conhecimento de pano de fundo e hipóteses. É um campo que considera qualquer tipo de linguagem de programação para representar hipóteses, como programações funcionais (PLOTKIN, 1970).

Um método utilizado para classificação e regressão é o método de máquinas de vetores de suporte. A partir de um conjunto de treinamento, marcado como pertencente a uma ou mais categorias, o algoritmo com este método constrói um modelo capaz de dizer em qual categoria um novo exemplo irá se encaixar. Esse método funciona muito bem em domínios “complicados”, que ocorrem quando apresentam uma margem significativa de pontos separados, porém não funciona bem em grandes conjuntos de dados e com muitos ruídos (GUNN, 1998).

O método clustering ou de agrupamento de dados realiza agrupamentos automáticos de dados de acordo com seu grau de semelhança. O critério dessa semelhança faz parte da definição do problema e depende do algoritmo. Normalmente, o usuário deve escolher, pelo menos, o número de grupos a serem detectados (SEGARAN, 2007).

As redes bayesianas são modelos baseados no teorema de Bayes. Matematicamente trata-se de uma tabela de conjuntos de probabilidades do universo do problema. Essas redes constituem um modelo que representa as relações de causalidade das variáveis de um sistema baseados em incerteza (KORB e NICHOLSON, 2003). Recentemente as redes bayesianas vem se tornando uma metodologia para construção de sistemas que confiam no conhecimento probabilístico e pode ser aplicada em muitos casos do mundo real (BOBBIO et al., 2001).

Tem também o algoritmo genético, o qual se trata de uma técnica de busca utilizada para encontrar soluções aproximadas em problemas de otimização e busca. Segundo Linden (2008), estes algoritmos são diferentes de algoritmos tradicionais de otimização devido a alguns aspectos: Se baseiam em uma codificação do conjunto das soluções possíveis; Os resultados são apresentados como múltiplas soluções; Não exigem um conhecimento prévio do problema e usam transições probabilísticas para a solução do problema.

### 2.1.2. Softwares com algoritmos de aprendizado de máquina

Na internet existe uma gama de softwares de aprendizados de máquina disponíveis para downloads. Diversos são gratuitos e outros são pagos. Ao adquirir um desses softwares, o usuário encontra alguns algoritmos de aprendizado de máquina e pode utilizá-los de acordo com o seu interesse. Neste tópico abordaremos alguns destes softwares.

#### 2.1.2.1. Auto Weka

Plataforma muito utilizada devido à sua interface de fácil utilização, possui diversos algoritmos. A versão mais atualizada utiliza otimização bayesiana para encontrar um algoritmo preciso de acordo com o conjunto de dados (THORNTON et al., 2013).

Sejam os algoritmos presentes no software  $\mathcal{A} = \{A^{(1)}, A^{(2)}, \dots, A^{(k)}\}$  e os seus espaços de hiperparâmetros associados  $\Lambda^{(1)}, \Lambda^{(2)}, \dots, \Lambda^{(k)}$ , ele tem como objetivo encontrar a combinação de algoritmo  $A^{(j)} \in \mathcal{A}$  e os hiperparâmetros  $\lambda \in \Lambda^{(j)}$  que minimizam a perda da validação cruzada, mostrada na equação a seguir (KOTHOFF et al., 2016):

$$A_{\lambda^*}^* = \underset{A^{(j)} \in \mathcal{A}, \lambda \in \Lambda^{(j)}}{\operatorname{argmin}} \frac{1}{k} \sum_{i=1}^k \mathcal{L}(A_{\lambda}^{(j)}, \mathcal{D}_{train}^{(i)}, \mathcal{D}_{test}^{(i)}) \quad (1)$$

Onde  $\mathcal{L}(A_{\lambda}^{(j)}, \mathcal{D}_{train}^{(i)}, \mathcal{D}_{test}^{(i)})$  representa a perda alcançada pelo algoritmo A com hiperparâmetros  $\lambda$ , treinado em  $\mathcal{D}_{train}^{(i)}$  e avaliado em  $\mathcal{D}_{test}^{(i)}$ .

Essa perda pode ser interpretada como um problema de otimização do tipo caixa preta, definido pela seguinte equação:

$$\operatorname{argmin}_{\theta \in \Theta} f(\theta) \quad (2)$$

Onde cada configuração  $\theta \in \Theta$  compreende a escolha do algoritmo  $A^{(j)} \in \mathcal{A}$  e os hiperparâmetros  $\lambda \in \Lambda^{(j)}$ .

A otimização bayesiana ou otimização baseada em modelo sequencial é uma ótima ferramenta para a resolução de problemas de otimização do tipo caixa preta (BROCHU et al., 2010).

Considerando a sua iteração de número  $n$ , ele se ajusta a um modelo probabilístico baseado nas primeiras  $n - 1$  avaliações de funções, como mostrado na Equação 3.

$$\langle \theta_i | f(\theta_i) \rangle_{i=1}^{n-1} \quad (3)$$

Este modelo é usado para selecionar o próximo  $\theta_n$  e avaliar  $f(\theta_i)$ . A otimização bayesiana é baseada modelos de árvores, pois de acordo com Thornton (2013), esta se torna mais eficiente quando aplicada em problemas complexos.

### 2.1.2.2. AutoGluon – Tabular

Erickson et al. (2020) desenvolveram o software AutoGluon-Tabular em cima de algumas características que julgaram ser importantes para um algoritmo de aprendizado de máquina, estas características são apresentadas a seguir:

- **Simplicidade:** Um usuário pode utilizar sem conhecer os detalhes sobre os dados e os modelos existentes no software;
- **Robustez:** A estrutura é capaz de lidar com uma grande variedade de conjunto de dados garantir a solução mesmo quando alguns modelos individuais falharem;
- **Tolerância ao erro:** O treinamento pode ser interrompido e retomado a qualquer momento;
- **Tempo previsível:** Retorna os resultados dentro do intervalo de tempo especificados pelos usuários.

O software funciona a partir da leitura de um arquivo .csv. Então, o AutoGluon processa os dados brutos identificando o tipo de problema, particiona os dados em vários grupos para treinamento do modelo e validação, faz o ajuste individual a vários modelos e cria um conjunto de modelos otimizados que supera os modelos individuais que foram treinados anteriormente. Alguns dos modelos presentes no AutoGluon são: redes neurais, árvores impulsionadas LightGBM e CadBoost, florestas aleatórias, árvores extremamente aleatórias e vizinhos mais próximos.

### **2.1.2.3. H2O AutoML**

Trata-se de um algoritmo automatizado simples de usar que produz resultados de qualidade adequados ao ambiente corporativo. Compreende modelos de regressão, classificação binária, classificação multiclasse, gradiente GBM, variedade de árvores extremamente randomizados, redes neurais profundas e modelos lineares generalizados. É bastante utilizado na indústria. (LEDELL e POIRIER, 2020).

O software funciona identificando quais hiperparâmetros são considerados mais importantes para cada algoritmo, além dos seus intervalos definidos. Utiliza ainda uma pesquisa aleatória para gerar modelos.

De acordo com Lendell e Poirier (2020), os modelos pré especificados são capazes de fornecer padrões rápidos e confiáveis para cada algoritmo. A ordem a qual o software utilizará cada algoritmo pode ser definida pelo usuário. Para isso, é importante que o usuário tenha um conhecimento acerca dos seus dados e dos modelos existentes no software.

Após treinar os modelos base, outros dois modelos chamados “stacked Ensemble” são utilizados para determinar a validação cruzada. Esta técnica se mostra muito eficiente (LAAN et al., 2007). A validação cruzada é desaconselhada quando os dados são muito grandes ou quando existe uma dependência de tempo entre linhas nos dados. A interface foi projetada para ter o menor número possível de parâmetros indicados pelo usuário, facilitando a sua utilização.

Para Lendell e Poirier (2020), os pontos positivos dessa ferramenta incluem a sua facilidade de utilização, escalabilidade para conjuntos grandes, e suporte a diferentes idiomas. Sendo assim, uma ferramenta que pode ser usada em diversas equipes de estatísticos, cientistas e engenheiros, tornando-se prática em qualquer equipe heterogênea.

#### **2.1.2.4. Sumo Toolbox**

Trata-se de um kit de ferramentas disponível em Matlab capaz de realizar ajuste de dados, seleção de modelos, seleção de amostras, otimização de hiperparâmetros e computação. Ele utiliza aproximações globais baseadas nos dados usando modelos substitutos compactos como funções racionais, kriging, redes neurais, splines e suporte de máquinas de vetores.

De acordo com Gorissen et al. (2010), a toolbox possui muitos plugins diferentes e cada componente do software é facilmente configurável através de um arquivo central e apresenta um suporte embutido para computação de alto desempenho.

O processo de geração de modelos consegue ter um tempo consideravelmente melhor do que outras ferramentas. As simulações podem ocorrer localmente ou em um cluster separado e a interface é de fácil utilização e totalmente automática.

Crombecq (2011) descreve o passo a passo de como a toolbox funciona, conforme mostrado a seguir.

A primeira etapa é definir o projeto inicial, o qual deve ser suficientemente grande para que seja possível utilizar a estratégia de design sequencial. A etapa seguinte e extremamente importante é o tipo do modelo. Na ferramenta, o conjunto de modelos substitutos ou metamodelos disponíveis são: Kriging, polinômios, função de base radial, splines, redes neurais artificiais e máquinas de vetor de suporte com mínimos quadrados. Existe ainda algoritmos de otimização disponíveis para cada um desses modelos.

Depois de gerado, o modelo deve ser validado por meio da ferramenta, que disponibiliza alguns tipos de validação tais como: validação cruzada, conjunto de validação, leave-one-out, diferença de modelos, entre outros.

A configuração da toolbox é feita por dois arquivos. O primeiro é a configuração do simulador, no qual deve-se determinar o número de entradas e saídas, os arquivos executáveis e o conjunto de dados associados com suas respectivas restrições. O segundo arquivo compreende a configuração global da toolbox, compreendendo os critérios de parada e o diretório de saída.

A estrutura modular da toolbox permite que o usuário experimente diferentes combinações de componentes para encontrar a melhor combinação para o problema. Caso o usuário não tenha muita experiência, existe uma configuração padrão pronta para ser usada, caso tenha, ele pode alterar as configurações para ajustar ainda mais o processo de escolha dos modelos.

#### **2.1.2.5. Metodologia ALAMO**

Metodologia capaz de aprender funções algébricas de dados desenvolvida por Cozad et al. (2014). A construção desta metodologia foi motivada pela necessidade de obter modelos simples a partir de dados de simulações.

A técnica ALAMO, conforme mostrada por Wilson e Sahidinis (2017), considera um grande número de transformações explícitas na variável de entrada  $x$ . O modelo é testado e melhorado usando otimização que mostram novos pontos de forma adaptativa.

Seja  $N^{ini}$  um conjunto de dados inicial. Um modelo substituto é então construído utilizando uma metodologia de maximização do erro (EMS), a qual é utilizada para identificar os próximos pontos  $i' = N + 1 \dots N^{ems}$ . Se o modelo selecionado satisfizer a tolerância do erro  $\delta$ , o algoritmo é autorizado a parar. Caso contrário, outros pontos são anexados ao conjunto de treinamento inicial.

Cozad, Sahidinis e Miller (2015) descreveram a metodologia ALAMO, conforme será apresentada a seguir. ALAMO desconhece a forma funcional de



um modelo de regressão. Assim, ele apresenta um conjunto simples de funções básicas, por exemplo,  $x, x^2, \frac{1}{x}, \log(x)$ , e um termo constante e, a partir disso, tenta construir uma função de baixa complexidade utilizando programação quadrática inteira mista (MIQP). Na solução do MIQP, as funções básicas  $X_j(x), j \in \beta$ , são ativas quando a variável correspondente binária for  $y_j = 1$  e inativa quando for  $y_j = 0$ . O tamanho do modelo é especificado por um parâmetro  $T$  correspondente ao número de funções ativas e este parâmetro vai aumentando até alcançar um critério de parada.

A partir da lista de funções básicas mostradas acima, o MIQP segue a seguinte equação:

$$\min g(\beta) = \sum_{i=1}^N \left( z_i - \left[ \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 \frac{1}{x} + \beta_4 \log(x) \right] \right)^2 \quad (4)$$

Onde:

$$\beta_j^{\min} y_j \leq \beta_j \leq \beta_j^{\max} y_j \quad j = 0, \dots, 4 \quad (5)$$

$$y_0 + y_1 + y_2 + y_3 + y_4 = T \quad (6)$$

$$y_j \in \{0,1\} \quad j = 0, \dots, 4 \quad (7)$$

Quando o modelo é identificado, ele é melhorado sistematicamente por meio de uma técnica de amostragem adaptativa que adiciona novos pontos, sejam de simulações ou experimentais.

## 2.2. Técnicas de Amostragem

Durante as pesquisas, tem-se uma grande quantidade de dados a serem analisados. É inviável, para não dizer impossível, que se analise todos os casos possíveis para se chegar a alguma conclusão, isso demandaria muito tempo e muito esforço computacional. Por isso, existem as técnicas de amostragem, que são capazes de selecionar algumas amostras que representem o conjunto total de possibilidades.

Escolher a técnica de amostragem nem sempre é uma tarefa fácil, existem inúmeros métodos que podem ser escolhidos e é importante que o pesquisador entenda-os para fazer a escolha correta no seu estudo.

Para Taherdoost (2016), as técnicas de amostragem podem ser divididas em dois tipos principais: Probabilidade ou amostragem aleatória; Amostragem probabilística ou não aleatória.

A amostragem aleatória significa que qualquer ponto tem igual probabilidade de ser escolhido para fazer parte da amostra a ser analisada, exemplo disso é utilizar um programa de computador que seja construído para fazer a seleção aleatória de dados (ZIKMUND, 2002).

A amostragem probabilística, ou não aleatória, tende a se concentrar em pequenas amostras destinadas a examinar um fenômeno específico e não para fazer inferências sobre o todo o conjunto inteiro de dados (YIN, 2003).

Quando se trata de construção de metamodelos, a etapa de amostragem é um item importante de análise pois ela visa selecionar pontos que serão avaliados por experimentos, e estes pontos servirão como base para a construção desses metamodelos (FRISSE, SCARPEL e FERRARI, 2011).

A técnica de espalhamento de amostras no espaço pode ser dividida em dois tipos: utilizando um número fixo de amostras ou utilizando um número adaptativo de amostras. Essas técnicas serão descritas no decorrer desta seção.

### **2.2.1. Técnicas de Amostragem para tamanho de amostras fixo**

#### **2.2.1.1. Análise Fatorial**

Técnica de amostragem bastante eficiente quando o estudo envolve a análise do efeito de dois ou mais fatores. Neste tipo de amostragem, todas as combinações possíveis dos níveis dos fatores são investigadas (MONTGOMERY, 2017).

Existem situações que envolvem um vasto número de variáveis a serem observadas. Entretanto, se entre essas variáveis existir variáveis correlacionadas, é possível agrupá-las de modo que variáveis pouco correlacionadas fiquem localizadas em grupos distintos (CARVALHO, 2013).

Para Hair et al. (2009), a análise fatorial é uma técnica de interdependência, com o objetivo de sintetizar a informação de diversas variáveis originais em um conjunto menor de novas dimensões.

De acordo com Pereira et al. (2019), a análise fatorial é adequada em quatro situações: Especificação da unidade de análise; Obtenção de resumo ou redução de dados; Seleção de variáveis; Uso dos resultados com outras técnicas.

Para Montgomery (2017), a análise fatorial possui algumas vantagens: É mais eficiente do que a análise de um fator por vez; É necessária quando as interações estão presentes e assim evitam conclusões enganosas; Permite que os efeitos de um fator sejam estimados em vários níveis, produzindo conclusões que são válidas em uma variedade de condições experimentais.

A notação utilizada para este método de amostragem traz uma gama de informações. Se o design experimental é do tipo  $2^3$ , significa que tem-se 3 fatores e cada fator tem 2 níveis, totalizando 8 condições experimentais diferentes. Um experimento fatorial pode ser analisado utilizando a análise de regressão ou a ANOVA (COHEN, 1968).

Observaremos um exemplo geral de um caso com dois fatores, conforme mostrado por Montgomery (2017). As observações podem ser descritas por um modelo, como mostrado na equação a seguir:

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} \begin{cases} i = 1, 2, \dots, a \\ j = 1, 2, \dots, b \\ k = 1, 2, \dots, n \end{cases} \quad (8)$$

Onde  $y_{ijk}$  é a resposta observada quando o fator A representa o  $i$ -ésimo nível ( $i = 1, 2, \dots, a$ ) e o fator B representa o  $j$ -ésimo nível ( $j = 1, 2, \dots, b$ ) para a  $k$ -ésima repetição ( $k = 1, 2, \dots, n$ );  $\mu$  é o efeito médio geral;  $\tau_i$  é o efeito do  $i$ -ésimo nível do fator de linha A,  $\beta_j$  é o efeito do  $j$ -ésimo nível do fator coluna B,  $(\tau\beta)_{ij}$  é efeito da interação entre  $\tau_i$  e  $\beta_j$ ;  $\varepsilon_{ijk}$  é um termo de erro aleatório.

Assume-se que os dois fatores são fixos e os efeitos do tratamento são definidos como desvios da média geral, assim:

$$\sum_{i=1}^a \tau_i = 0 \quad (9)$$

E,

$$\sum_{j=1}^b \beta_j = 0 \quad (10)$$

De forma análoga, os efeitos das interações são fixos e definidos conforme a equação a seguir:

$$\sum_{i=1}^a (\tau\beta)_{ij} = \sum_{j=1}^b (\tau\beta)_{ij} = 0 \quad (11)$$

Em uma análise fatorial com dois fatores, os fatores A e B são tratados com igual interesse e existe um interesse em testar hipóteses sobre a igualdade dos efeitos linha:

$$\begin{cases} H_0: \tau_1 = \tau_2 = \dots \tau_a = 0 \\ H_1: \text{Pelo menos um } \tau_i \neq 0 \end{cases} \quad (12)$$

A igualdade dos efeitos coluna:

$$\begin{cases} H_0: \beta_1 = \beta_2 = \dots \beta_b = 0 \\ H_1: \text{Pelo menos um } \beta_j \neq 0 \end{cases} \quad (13)$$

Também é importante testar a interação entre os tratamentos:

$$\begin{cases} H_0: (\tau\beta)_{ij} = 0 \text{ para todo } i, j \\ H_1: \text{Pelo menos um } (\tau\beta)_{ij} \neq 0 \end{cases} \quad (14)$$

### **Design ideal (D-optimal design)**

Este método de amostragem corresponde a uma classe do design de experimentos considerado ótimo em relação a alguns critérios estatísticos. Este método permite que alguns parâmetros sejam estimados sem viés e com uma variância mínima, além de reduzir os custos experimentais.

De acordo com Atkinson, Donev e Tobias (2007), existem algumas vantagens neste tipo de método: A redução nos custos experimentais; A capacidade de acomodação de vários tipos de fatores; Os designs que podem ser otimizados quando o espaço de processo matemático contém configurações de fator que são praticamente inviáveis.

Existem vários critérios estatísticos para avaliar este método, um dos mais importantes é o critério "D-ótimo". Este critério é bastante utilizado quando a região experimental não apresenta regularidades, quando o número de

experimentos em técnicas clássicas de amostragem é muito alto ou quando se tem a intenção de aplicar modelos diferentes dos modelos de primeira e segunda ordem (DE AGUIAR et al., 1995).

Este critério foi inicialmente proposto por Smith (1918) e consiste em encontrar uma matriz ótima contendo os pontos candidatos dentre os inúmeros pontos possíveis seguindo um critério a partir do determinante da matriz de dispersão.

Seja  $\xi_N$  a matriz que contém todos os pontos candidatos,  $X$  a matriz com os coeficientes do modelo e  $X^*$  a matriz ótima que contém os melhores pontos a serem estimados. O critério-D pode ser alcançado pela seguinte equação.

$$\det(X^*{}'X^*)^{-1} = \min_{\xi_n \Xi_n} (\det(X'X)^{-1}) \quad (15)$$

Onde  $\xi_n \Xi_n$  representa o grupo de todas as matrizes  $\xi_n$ , escolhidas a partir de  $\xi_N$ ;

### 2.2.1.2. O conceito de discrepância

Muitas técnicas de amostragem se baseiam na construção de amostras com baixa discrepância entre elas. O conceito de discrepância, de acordo com Tezuka (1994), é mostrado na equação a seguir.

$$D_N^{(k)} = \sup_J \left| \frac{A(J;N)}{N} - V(J) \right| \quad (16)$$

Onde  $N$  corresponde aos pontos  $X_0, X_1, \dots, X_{N-1}$  em  $[0,1]^k, k \geq 1$ ;  $J$  corresponde a um subintervalo  $J = \prod_{i=1}^k [0, u_i], k \geq 1$ , onde  $0 < u_i \leq 1$  para  $1 \leq i \leq k$ ;  $A(J; N)$  corresponde ao número de  $n$ , onde  $n$  pertence ao intervalo  $0 \leq n < N$ , com  $X_n \in J$ ;  $V(J)$  representa o volume de  $J$ ; o termo *sup* representa que o intervalo explicitado abrange todos os subintervalos de  $J$ .

E uma sequência com baixa discrepância é representada pela equação, para todo  $N > 1$ :

$$D_N^{(k)} \leq C_k (\log N)^k / N \quad (17)$$

Onde  $C_k$  é uma constante que depende apenas da dimensão  $k$ .

### 2.2.1.3. Sequências de Sobol

As sequências de Sobol representam um exemplo de sequência de baixa discrepância considerada quase aleatória. De acordo com Joe e Kuo (2003), as sequências de Sobol utilizam uma base dois para formar partições uniformes sucessivamente mais finas no intervalo de unidade e, após feito isso, reordenam as coordenadas em cada dimensão.

Publicado inicialmente por Sobol (1967), esta técnica consiste em encontrar boas distribuições na unidade do hipercubo s-dimensional.

Seja a unidade do hipercubo s-dimensional  $I^S = [0,1]^S$ , e f uma função possível de ser integrada sobre  $I^S$ , o objetivo então consiste em construir uma sequência  $X_n$  em  $I^S$  que respeite a equação a seguir e que a convergência seja a mais rápida possível:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i) = \int_{I^S} f \quad (18)$$

Por ser um método bastante útil, o software MATLAB® possui uma função chamada “sobolset” capaz de gerar os pontos utilizando a técnica de Sobol. A função padrão segue a implementação proposta por Joe e Fuo (2003), nela o objeto “p” contém um conjunto de pontos com d-dimensões. Cada ponto  $p(i,:)$  corresponde a um ponto na sequência de Sobol. A j-ésima coordenada do i-ésimo ponto  $p(i,j)$  é então definida como:

$$\begin{cases} 0, & i = 1 \\ \gamma_i(1)v_j(1) \oplus \gamma_i(2)v_j(2) \oplus \dots, & i > 1 \end{cases} \quad (19)$$

Onde os valores do parâmetro  $\gamma_i(n)$  são 0 ou 1 de forma que:

$$i - 1 = \sum_{n=1} \gamma_i(n) 2^{n-1} \quad (20)$$

Ou seja,  $\gamma_i(n)$  representa os dígitos binários do inteiro  $i-1$ .

Onde  $\oplus$  é um operador exclusivo. Para dois números expressos em binários, este operador compara os dígitos em cada posição. O operador retorna o valor 1 se o dígito difere naquela posição e retorna 0 se o dígito é igual naquela posição.

Os valores  $v_j(n)$  representam a seguinte equação e dependem exclusivamente da coordenada j:

$$v_j(n) = \frac{m_j(n)}{2^n} \quad (21)$$

#### 2.2.1.4. Sequências de Halton

Considerada uma das sequências com baixa discrepância mais popular, é utilizada para gerar pontos no espaço para métodos numéricos e é considerada uma sequência quase aleatória (Schlier, 2008). As sequências de Halton utilizam diferentes bases principais em cada dimensão para preencher o espaço de maneira uniforme.

De acordo com Drukker e Gates (2006), a geração de pontos pela sequência de Halton acontece da seguinte forma. Considerando um número inteiro não negativo  $i$ , este apresenta uma expansão do tipo:

$$i = \sum_{j=1}^q b_{j,p}(i) p^{j-1} \quad (22)$$

Onde  $p^{j-1} \leq i < p^q$ ;  $p$  é restrito a ser um número primo;  $b_{j,p}(i) \in \{0, 1, 2, \dots, p-1\}$  corresponde ao  $j$ -ésimo dígito de base  $p$ .

O  $i$ -ésimo dígito é obtido pela seguinte equação:

$$r_p(i) = \sum_{j=1}^q \frac{b_{j,p}(i)}{p^j} \quad (23)$$

No software Matlab existe uma função para a geração de pontos segundo a sequência de Halton, chamada "haltonset". De acordo com Kocis e Whiten (1997), o default desta função contém um objeto "p" com pontos de  $d$ -dimensões. Cada  $p(i,:)$  corresponde a um ponto da sequência de Halton. A coordenada " $j$ " do ponto é dada pela equação:

$$\sum_k a_{ij}(k) b_j^{-k-1} \quad (24)$$

Onde  $b_j$  corresponde ao  $j$ -ésimo número; Os coeficientes  $a_{ij}(k)$  são inteiros não negativos menores do que  $b_j$ , tal que:

$$i - 1 = \sum_{k=0} a_{ij}(k) b_j^k \quad (25)$$

### 2.2.1.5. Sequências de Hammersley

De acordo com Ke et al. (2012), a sequência de Hammersley se baseia na teoria que cada inteiro não negativo  $k$  pode ser expandido de acordo com a seguinte equação:

$$k = a_0 + a_1p + a_2p^2 + \dots + a_r p^r, a \in [0, p - 1] \quad (26)$$

Considerando um problema de dimensão  $d$ , a base principal  $p$  é escolhida como uma sequência:  $p_1, p_2, \dots, p_{d-1}$ , estes termos definem a sequência de funções  $\Phi_{p_1}, \Phi_{p_2}, \dots, \Phi_{p_{d-1}}$ . E a função  $\Phi_p(k)$  é definida por:

$$\Phi_p(k) = \frac{a_0}{p} + \frac{a_1}{p^2} + \frac{a_2}{p^3} + \dots + \frac{a_r}{p^{r+1}} \quad (27)$$

Portanto, o  $k$ -ésimo ponto de dimensão  $d$  é dado por:

$$\left( \frac{k}{n}, \Phi_{p_1}(k), \Phi_{p_2}(k), \dots, \Phi_{p_{d-1}}(k) \right) \quad \text{para } k = 0, 1, 2, \dots, n - 1 \quad (28)$$

Onde  $n$  é igual ao número de pontos amostrais e  $p_1 < p_2 < \dots < p_{d-1}$ .

### 2.2.1.6. Voronoi

O diagrama de Voronoi corresponde a uma partição do plano em regiões próximas. Em casos simples, esses objetos são pontos finitos, chamados sementes e, para cada semente, existe uma região correspondente denominada células de Voronoi. As células de Voronoi consistem em espaços que possuem os pontos mais próximos de uma semente do que de outra semente (AURENHAMMER, 1991).

Em casos mais simples, um conjunto de pontos  $\{z_1, z_2, \dots, z_n\}$  onde cada ponto  $z_k$  pertence a uma célula de Voronoi  $V_k$ . Estas células de Voronoi comportam todos os pontos no plano cuja distância até  $z_k$  é menor ou igual à distância deste mesmo ponto até um  $z_k$  de uma outra célula de Voronoi (BOYD e VANDENBERGUE, 2004).

Conforme mostrado por Okabe, Boots e Sugihara (1992), a distribuição dos pontos seguindo a amostragem Voronoi acontece da seguinte forma. Para um conjunto aberto  $\Omega \subseteq \mathbb{R}^N$ , o conjunto  $\{V_i\}_{i=1}^k$  é chamado de diagrama de Voronoi de  $\Omega$ , se  $V_i \cap V_j = \emptyset$  para  $i \neq j$  e  $\cup_{i=1}^k \bar{V}_i = \bar{\Omega}$ .



Dado um conjunto de pontos  $\{z_i\}_{i=1}^k \in \bar{\Omega}$ , a célula de Voronoi  $\hat{V}_i$  correspondente ao ponto  $z_i$  é definida pela seguinte equação:

$$\hat{V}_i = \{x \in \Omega \mid |x - z_i| < |x - z_j| \quad \text{para } j = 1, \dots, k, \quad j \neq i \quad (29)$$

O trabalho desenvolvido por Du, Faber e Gunzberg (1999) apresenta uma abordagem de amostragem de Voronoi onde os pontos espalhados no diagrama ocuparão a região central das células de Voronoi, conforme mostrado a seguir.

Seja um conjunto de pontos discretos  $W = \{y_i\}_{i=1}^m \in \mathbb{R}^N$ . O conjunto  $\{V_i\}_{i=1}^k$  corresponde a um diagrama de Voronoi de  $W$  se  $V_i \cap V_j = \emptyset$  para  $i \neq j$  e  $\cup_{i=1}^k \bar{V}_i = W$ .

Considerando um conjunto de pontos  $\{z_i\}_{i=1}^k \in \mathbb{R}^N$ . As células de Voronoi são definidas pela seguinte expressão:

$$\hat{V}_i = \{x \in W \mid |x - z_i| \leq |x - z_j| \quad \text{para } j = 1, \dots, k, \quad j \neq i \quad (30)$$

Ainda de acordo com Du, Faber e Gunzberg (1999), uma outra abordagem para o tratamento de pontos equidistantes é mostrada a seguir.

Seja uma função de densidade  $\rho$  definida em  $W$ , o centróide de massa  $z^*$  de um conjunto  $V \subset W$  é definido pela equação a seguir:

$$\sum_{y \in V} \rho(y) |y - z^*|^2 = \inf_{z \in V^*} \sum_{y \in V} \rho(y) |y - z|^2 \quad (31)$$

Onde as somas se estendem sobre os pontos pertencentes a  $V$  e  $V^*$  pode ser o conjunto  $V$  ou maior, como por exemplo, o conjunto  $\mathbb{R}^N$ .

### 2.2.1.7. Niederreiter

Seguindo as técnicas de amostragem com baixa discrepância, Harald Niederreiter, propôs uma técnica de construção geral para as sequências do tipo  $(t, k)$ , conforme Tezuka (1994) apresenta em seu estudo.

Seja  $t$ ,  $0 \leq t \leq m$  um número inteiro. Uma sequência  $(t, k)$ , na base  $b$ , é uma sequência de pontos em  $[0, 1]^k$  se para todos os inteiros  $j \geq 0$  e  $m > t$ , o conjunto de pontos  $[X_n]_m$  com  $jb^m \leq n < (j+1)b^m$  for do tipo  $(t, m, k)$  na base  $b$ .

Seja  $k \geq 1$ ,  $b \geq 2$  e  $B = \{0, 1, \dots, b-1\}$ . Para  $n = 0, 1, 2, \dots$ . Seja  $n = \sum_{r=1}^{\infty} a_r(n) b^{r-1}$  para  $a_r(n) \in B$ . A  $h$ -ésima coordenada do ponto  $X_n$  é dada pela seguinte equação:

$$X_n^{(h)} = \sum_{i=1}^{\infty} x_{ni}^{(h)} b^{-i} \quad (32)$$

Para  $1 \leq h \leq k$  e  $0 \leq n$ .

Onde:

$$x_{ni}^{(h)} = \lambda_{hi} \left( \sum_{j=1}^{\infty} c_{ij}^{(h)} \psi_j(a_j(n)) \right) \in B \quad (33)$$

Para  $1 \leq h \leq k, 1 \leq i$  e  $0 \leq n$ .

Onde:  $\psi_j: B \rightarrow R$  para  $j = 1, 2, \dots$ , com  $\psi_j = 0$  para todo  $j$  suficientemente grande;  $\lambda_{hi}: R \rightarrow B$  para  $h = 1, 2, \dots, k$  e  $i = 1, 2, \dots$ , com  $\lambda_{hi} = 0$  para  $1 \leq h \leq k$  e  $i$  suficientemente grande.

Denomina-se  $C^{(h)} = (c_{jr}^{(h)})$  a matriz geradora da  $h$ -ésima coordenada de uma sequência do tipo  $(t, k)$ .

### 2.2.1.8. Sequências de Faure

As sequências de Faure são semelhantes às sequências de Halton, mostradas anteriormente na seção 2.2.5. Entretanto, algumas diferenças são notadas, conforme apresentado por Yu, Goos e Vanderbroek (2010):

- As sequências de Faure utilizam apenas uma base para todas as dimensões, enquanto as sequências de Halton utilizam bases diferentes para cada dimensão;
- As dimensões superiores são geradas a partir de elementos das dimensões inferiores, por outro lado, as sequências de Halton emparelha  $p$  sequências unidimensionais.

Seguindo a construção da matriz geradora da  $h$ -ésima coordenada de uma sequência  $(t, k)$ , mostrada na seção 2.2.8. A geração da matriz da sequência de Faure, de acordo com Tezuka (1994), é dada pela equação:

$$C^{(h)} = A^{(h)} p^{h-1} \quad (34)$$

Onde  $1 \leq h \leq k$ ;  $A^{(h)} = I$  para todo  $h$ , considerando a sequência de Faure original.

### 2.2.1.9. Latin Hiper cubo

O Latin Hiper cubo foi apresentado pela primeira vez em 1979 por McKay, Beckman e Conover. Esta técnica divide o domínio de cada variável aleatória em faixas (OLSSON, SANDEBERG e DAHLBLOM, 2003), cada faixa é amostrada uma única vez, resultando em uma distribuição esparsa dos pontos, o que garante uma cobertura homogênea do domínio das variáveis aleatórias (SANTOS, 2014).

Seja  $nv$  o número de variáveis aleatórias do problema e  $n$  o número de pontos da amostra. Uma matriz  $P$ , de dimensões  $n \times nv$  é gerada, onde cada uma das  $nv$  colunas é uma permutação aleatória de 1 até  $n$ . Outra matriz  $R$  é gerada com as mesmas dimensões, cujos componentes são números aleatoriamente distribuídos entre  $(0,1)$ . Então é possível obter uma matriz  $S$  a partir da seguinte equação (OLSSON, SANDEBERG e DAHLBLOM, 2003):

$$S = \frac{1}{n}(P - R) \quad (35)$$

As amostras são geradas a partir de  $S$ , tal que:

$$x_{ij} = F_{x_j}^{-1}(s_{ij}) \quad (36)$$

Onde  $F_{x_j}^{-1}$  é a inversa da função de distribuição acumulada de probabilidade da variável  $X_j$ .

De acordo com Dehrendorff (2010), o método de amostragem LHS (*Latin Hypercube Sampling*) segue algumas propriedades:

- Os pontos são escolhidos aleatoriamente, mas não de uma forma independente;
- A média é enviesada;
- Cada variável é dividida em  $n$  estratos com igual probabilidade marginal.

### 2.2.1.10. Máxima Entropia

O método de amostragem por meio da técnica de máxima entropia é um critério construtivo para configurar uma distribuição de probabilidade geralmente declarado como restrições em valores de expectativa de algumas funções (COSSIO e DIAZ).

Sebastiani e Wynn (2000) apresentam a descrição da amostragem por máxima entropia. Esta descrição é apresentada a seguir.

Seja  $Y$  um  $n$ -vetor aleatório em  $y$  e  $\theta$  um  $p$ -vetor aleatório em  $\Omega$  e dado  $\Theta = \theta$  e um experimento  $\xi$ , então,  $Y$  possui uma distribuição conhecida com densidade de probabilidade  $p(y|\theta, \xi)$ . Supondo que  $\theta$  tenha uma distribuição com densidade de probabilidade  $p(\theta)$  que é independente de  $\xi$ , então, dado  $\xi$ , o par  $(Y, \theta)$  terá uma distribuição em  $y \times \Omega$ . Supondo ainda que  $\xi$  será escolhido a partir de um conjunto de experimentos  $\Xi$  para conseguir a máxima quantidade de informação de  $\theta$ , o ganho esperado de  $\xi$  sobre a distribuição de  $Y$  é dado por:

$$Ent(\theta) = E_Y\{Ent(\theta|Y, \xi)\} \quad (39)$$

Ainda de acordo com Sebastiani e Wynn (2000), a amostragem pela máxima entropia sugere dois teoremas:

- i) Supondo que o objetivo do experimento seja adquirir o máximo de informações sobre  $\theta$  e sabendo que a entropia da distribuição de  $Y, \theta|\xi$  é limitada e independente de  $\xi$  e ainda que  $Ent(Y|\xi)$  e  $E_Y\{Ent(\theta|Y, \xi)\}$  são limitadas, então, um experimento que maximize a entropia da distribuição de  $Y$  será mais informativo para  $\theta$ .
- ii) O Teorema i é válido sempre que  $Ent(\theta|Y, \xi)$  não depender do planejamento de  $\xi$ .

### 2.2.1.11. Minimax e Maximin

Para Johnson, Moore e Ylvisaker (1990), a utilização desta técnica se destina ao uso no problema de seleção de locais quando a superfície é modelada

por uma distribuição anterior e as observações são feitas sem erros. Em seu trabalho, os três pesquisadores mostram as definições matemáticas da técnica, a qual é mostrada a seguir.

Considerando que  $T$  seja um conjunto e que exista uma função não negativa  $d$  em  $T \times T$ , então para todo  $s$  e  $t$  em  $T$ :

$$d(s, t) = d(t, s) \quad (40)$$

E,

$$d(s, t) \geq 0 \quad (41)$$

Se e somente se  $s = t$ . Ou ainda:

$$d(s, t) \leq d(s, u) + d(u, y) \quad (42)$$

Para todo  $s, t$  e  $u$  em  $T$ .

Considerando um subconjunto  $S \in T$ , com  $\text{card}(S) = n$ , sendo  $n$  fixo. Denomina-se  $S^*$  uma distância “minimax” projetada se:

$$\min_s \max_{t \in T} d(t, S) = \max_{t \in T} d(t, S^*) = d^* \quad (43)$$

Onde  $d(t, S) = \min_{s \in S} d(t, s)$ .

Considerando um subconjunto  $S \in T$ , com  $\text{card}(S) = n$ , sendo  $n$  fixo. Denomina-se  $S^o$  uma distância “maximin” projetada se:

$$\max_S \min_{s, s' \in S} d(s, s') = \min_{s, s' \in S} d(s, s') = d^o \quad (44)$$

### 2.2.1.12. Amostragem Experimental Uniforme (UD)

A amostragem Experimental Uniforme caracteriza-se como um tipo de amostragem que pode ser usada para experimentos computacionais e industriais; em situações que o modelo é, ainda, desconhecido. Esta técnica propõe que os pontos sejam espalhados uniformemente no domínio do experimento (FANG e LIN, 2003).

Conforme mostrado por Fang e Lin (2003), a notação utilizada para esta técnica de amostragem é do tipo  $U_n(q^s)$ , onde  $U$  significa a técnica,  $n$  representa

o número de experimentos,  $s$  o número de fatores e  $q$  o número de níveis. Para a implementação desta técnica é necessário seguir algumas etapas, as quais são mostradas a seguir.

- 1) Definição do domínio do experimento, escolha dos fatores e determinação do número de níveis em cada fator;
- 2) Escolha de uma tabela adequada para acomodar os fatores e os níveis;
- 3) A partir da tabela de amostragem uniforme, determina-se aleatoriamente a ordem a qual os experimentos serão executados;
- 4) Busca por um modelo que se adeque aos dados;

A técnica de Amostragem experimental Uniforme (UD) busca espalhar, como o nome sugere, uniformemente, os pontos no domínio. Supondo que existam  $s$  fatores em um experimento, sem perda de generalidade pode-se assumir que o domínio experimental é do tipo  $C^S = [0,1]^S$ . O objetivo, então, é selecionar um conjunto de  $n$  experimentos com pontos  $\mathcal{P} = \{x_1, \dots, x_n\} \subset C^S$  uniformemente espalhados em  $C^S$ .

Seja  $M$  uma medida de uniformidade de  $\mathcal{P}$  de tal forma que o menor  $M$  corresponde a uma melhor uniformidade. Seja  $\mathcal{Z}(n, s)$  o conjunto de  $n$  pontos em  $C^S$ , um conjunto  $\mathcal{P}^* \in \mathcal{Z}(n, s)$  é chamado de amostra uniforme se houver um valor mínimo de  $M$  sobre  $\mathcal{Z}(n, s)$ :

$$M(\mathcal{P}^*) = \min_{\mathcal{P} \in \mathcal{Z}(n, s)} M(\mathcal{P}) \quad (45)$$

As três observações seguintes representam a chave para a construção dos UD:

- 1) Definir uma medida adequada de uniformidade;
- 2) Reduzir a complexidade do cálculo de pesquisa de UD;
- 3) Aplicar um algoritmo de otimização para encontrar o melhor UD.

### 2.2.2. Técnicas de Amostragem Sequencial

As técnicas de amostragem sequencial transformam o espalhamento de dados em um processo iterativo. Estas técnicas analisam dados de iterações

prévias, para que a seleção de novas amostras sejam espalhadas em regiões mais difíceis (CROMBECQ, 2011). São utilizadas para melhorar um design inicial como um Latin Hipercubo dando ênfase em partes altamente dinâmicas do espaço (CROMBECQ et al., 2009).

De acordo com Crombecq, Laermans e Dhaene (2011), alguns critérios importantes para a escolha da técnica adequada devem ser elencados como: Granularidade, onde a estratégia é considerada refinada se o método é capaz de selecionar um pequeno número de pontos a cada iteração do algoritmo; Preenchimento do espaço; Boas propriedades, por exemplo, considera-se um método com boas propriedades quando não existe a possibilidade de projeção de pontos no espaço, ou seja, dois pontos não ocupam o mesmo lugar neste espaço amostral; Ortogonalidade.

Para Crombecq (2011), o planejamento fatorial é a técnica de amostragem mais simples em relação ao preenchimento do espaço amostral, considerando número de amostras fixos. Ela maximiza a distância entre locais para cada número de  $m^d$  pontos.

Entretanto, ainda de acordo com Crombecq (2011), existem muitas desvantagens como:

- O método não é refinado, o fatorial só pode ser definido para a  $d$ -ésima potência de um inteiro  $m$ , o qual deve ser determinado antecipadamente. Para um refinamento mais aprimorado, seria necessário aumentar o planejamento em aproximadamente  $2^d$  para cada iteração.
- Não possui boas propriedades de projeto, se um parâmetro não for importante, cada um dos pontos desse parâmetro será avaliado  $m$  vezes.

A técnica de distribuição de pontos do Latin hipercubo é muito popular por apresentar uma matemática de fácil compreensão e fácil de ser implementada. Entretanto, alguns algoritmos que utilizam esta técnica não possuem boas propriedades de preenchimento no espaço (CROMBECQ, 2011). Um grande impasse para a utilização desta técnica é que bons algoritmos

demandam um esforço computacional muito grande, e então, se tornam inviáveis.

### 2.2.2.1. Lola-Voronoi

Técnica de amostragem proposta por Crombecq (2008), a qual utiliza um critério de exploração baseado em aproximações lineares locais do sistema (Lola) e um critério de exploração usando um mosaico de Voronoi.

O componente Lola indica que a amostragem deve ser proporcional à linearidade local da função, são necessárias menos amostras em regiões onde o sistema é quase linear e mais amostras em regiões menos lineares. Esta linearidade pode ser estimada calculando uma aproximação linear local em cada amostra.

A melhor aproximação linear local de uma função  $f$  é o gradiente desta função, conforme apresentado por Crombecq et al. (2008):

$$\nabla f = \left( \frac{\partial f}{\partial x^1}, \frac{\partial f}{\partial x^2}, \dots, \frac{\partial f}{\partial x^d} \right) \quad (46)$$

O gradiente deverá ser estimado pois a derivada da função é raramente conhecida. Entretanto, não se pode utilizar métodos tradicionais de estimativa de gradiente porque não há como afirmar que os pontos estão espalhados uniformemente. Assim, a estimativa do gradiente é obtida a partir de uma regressão dos mínimos quadrados aplicada às amostras na vizinhança (FU 2005).

O componente Voronoi é usado para estimar a densidade das amostras por meio de uma aproximação da tesselação de Voronoi. Quando uma amostra tem um grande tamanho de célula de Voronoi, comparado com as outras, a região pode ser subamostrada e investigada com mais atenção (CROMBECQ et al., 2008).

Ainda de acordo com Crombecq et al. (2008), tendo o erro Lola e o tamanho relativo da célula de Voronoi, todos os pontos a serem espalhados são calculados. Uma quantidade maior de pontos ficará situada nas regiões não lineares e novas amostras são então selecionadas nessas regiões.



### 2.2.2.2. *SED Toolbox*

No software MATLAB® existe uma toolbox bastante eficaz no projeto de experimentos sequencial, chamada SED. Ela fornece ao usuário alguns algoritmos que geram um projeto experimental sequencial, com rapidez e fácil utilização.

A distância entre os locais é a menor distância entre dois pontos no espaço de amostragem. Assim, esse valor deve ser o mais alto possível para que os pontos sejam distribuídos uniformemente. A distância projetada também é importante pois corresponde à menor distância entre todos os pontos depois de terem sido projetados em um eixo do espaço. Essa distância também deve ser maximizada (SUMOWIKI).

Todos os algoritmos na toolbox foram otimizados e são eficientes tanto na distância entre locais quanto na distância projetada.

### 2.2.2.3. *Intersite-projected*

Neste projeto de experimento sequencial, é realizada a repetição de amostragem aleatória. De acordo com Bhattacharyya (2018), geralmente um grande número de pontos amostrais são gerados. Assim, é necessário calcular a distância entre locais, projetada como critério de julgamento de escolha da melhor amostra.

O problema de otimização é resolvido com uma única função objetivo, mostrada na equação a seguir:

$$dist(P, p) = \frac{(n+1)^{\frac{1}{d}-1}}{2} \min_{p_i \in P} \sqrt{\sum_{k=1}^d |p_i^k - p^k|^2} + \frac{n+1}{2} \min_{p_i \in P} \|p_i - p\|_{-\infty} \quad (47)$$

Conforme mostrado por Bhattacharyya (2018), a função objetivo calcula a distância entre locais do novo ponto em relação a todos os outros pontos já existentes. Neste método, a cada iteração é necessário gerar o número de  $kn$  amostras aleatórias, onde  $k$  é um parâmetro do algoritmo e  $n$  é o número total de amostras geradas até aquela iteração. Quanto maior for o valor de  $k$ , melhor será a qualidade do projeto experimental.

### 2.2.2.4. *Intersite-projected-threshold*

Esta outra proposta considera a distância projetada como a função de limite. A distância mínima projetada é especificada e os pontos da amostra menores do que esta distância mínima são descartados. Essa distância mínima é apresentada na equação a seguir, conforme mostrada por Bhattacharyya (2018).

$$d_{min} = \frac{2\alpha}{n} \quad (48)$$

Onde  $\alpha$  representa a restrição de tolerância. Se  $\alpha = 0$ , não haverá qualquer restrição e a distância projetada não será levada em consideração, se  $\alpha = 1$ , todos os pontos serão descartados conforme as amostras são distribuídas de forma aleatória. Ainda para Bathacharyya (2008), é possível obter bons resultados utilizando  $\alpha = 0,5$ .

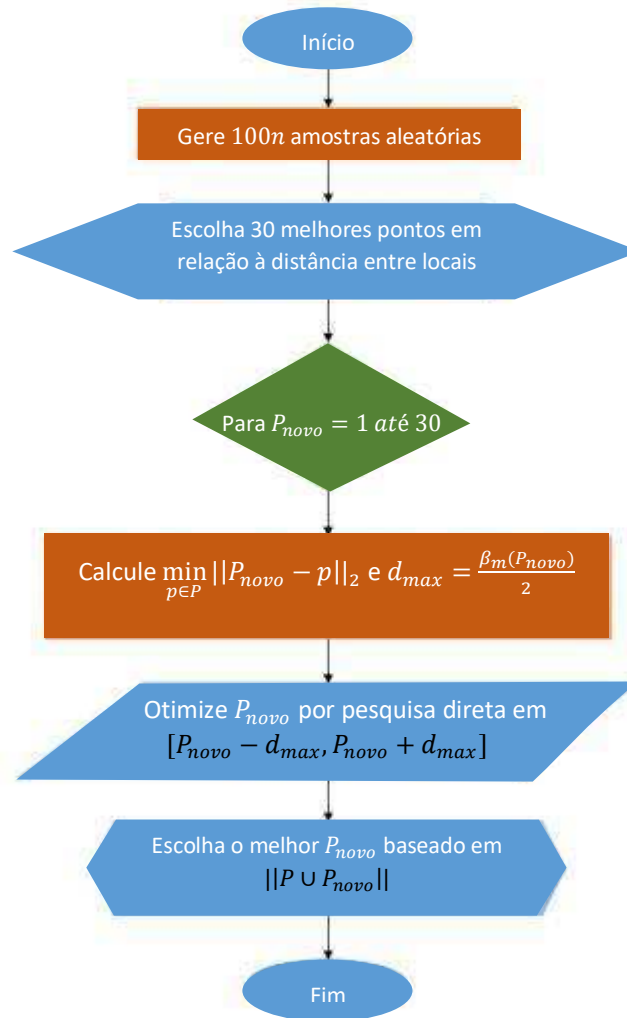
A função objetivo para esta técnica, é então mostrada na equação a seguir:

$$dis(P, p) = \begin{cases} 0 & , se \min_{p_i \in P} \|p_i - p\|_{-\infty} < d_{min} \\ \min_{p_i \in P} \|p_i - p\|_2 & , se \min_{p_i \in P} \|p_i - p\|_{-\infty} \geq d_{min} \end{cases} \quad (49)$$

#### **2.2.2.5. Optimizer-projected**

É possível melhorar o projeto experimental a partir da uso de uma otimização local. Assim, os pontos candidatos são otimizados em relação à distância projetada depois de ter sido feita a escolha dos melhores candidatos com relação à distância entre locais (BHATTACHARYYA, 2018).

A sequência de cálculo do algoritmo é apresentada por Bhattacharyya (2018), na Figura 1 mostrada a seguir:



**Figura 1: Sequência de cálculo do algoritmo**  
**Fonte: Bhattacharyya (2008)**

O fator  $\beta_m$  corresponde ao desvio dos pontos da sua localização para a otimização. Se  $\beta_m = 0$ , então significa que o projeto é baseado apenas na distância entre locais, não tendo influência da distância projetada. Se  $\beta_m = 1$ , significa que o problema estará completamente voltado para a otimização. Para Bhattacharyya (2008), um valor de  $\beta_m = 0,3$  torna o projeto de experimentos equilibrado entre os aspectos de distância entre os locais e distância projetada.

#### **2.2.2.6. Optimizer-intersite**

Nesta técnica, o ponto ótimo pode ser derivado das amostras sem a necessidade da utilização de um algoritmo específico para otimização. Considerando os intervalos criados, o ponto com a melhor distância projetada será o ponto no meio do hiper-cubo definido pelo maior intervalo em cada dimensão. Com os pontos gerados, uma busca padrão é realizada nos 50

maiores hipercubos, otimizando a distância entre locais (CROMBECQ, LAREMANS e DHAENE, 2011).

### 2.3. Transformações de variáveis dependentes e independentes

O método de transformação de variáveis pode ser bastante útil para reduzir a variância dos dados, e assim, conseguir alcançar os melhores modelos com um menor esforço computacional, e conseqüentemente, reduzindo o tempo de busca do modelo.

Alguns softwares, como o Minitab®, trazem esse artifício em seus algoritmos. De acordo com Suppor Minitab (2020), material disponibilizado pelo software, a transformação de variáveis se torna necessária quando os resíduos exibem uma variância não constante ou quando o modelo exibe falta de ajuste significativa, importante para a análise dos experimentos da superfície de resposta.

Se a transformação da variável for suficiente, é possível utilizar até mesmo uma análise de regressão para avaliar o conjunto de dados. Allaman (2018) disponibiliza alguns tipos de transformações mais comuns, as quais veremos a seguir.

A primeira delas é a transformação da raiz quadrada. Essa transformação é utilizada quando os dados são de contagem com lei de distribuição de poisson. Seja uma variável aleatória mensurada  $Y$ , tem-se a seguinte transformação:

$$Y' = \sqrt{Y} \quad (50)$$

Se houver zeros nos dados, uma dica é acrescentar um termo, conforme mostrado na equação a seguir:

$$Y' = \sqrt{Y + 0,5} \quad (51)$$

Uma outra transformação bastante utilizada é a transformação logarítmica que é indicada a sua utilização quando, apesar de ter dados contínuos, os mesmos não se aderem a uma distribuição normal. Seja uma variável aleatória mensurada  $Y$ , tem-se a seguinte transformação:

$$Y' = \log(Y) \quad (52)$$

Ou ainda,

$$Y' = \ln(Y) \quad (53)$$

Se houver zeros nos dados, uma dica é acrescentar um termo, conforme mostrado na equação a seguir:

$$Y' = \log(Y + 0,5) \quad (54)$$

Ou ainda,

$$Y' = \ln(Y + 0,5) \quad (55)$$

Uma outra transformação utilizada é a do acrcoseno da raiz quadrada. A função arcoseno é a inversa da função seno com domínio no intervalo  $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$  e imagem no intervalo  $[-1,1]$ . É aconselhável quando os dados são binomiais, apresentam proporções ou porcentagens. Seja uma variável aleatória mensurada  $Y$ , tem-se a seguinte transformação:

$$Y' = \arcsen\sqrt{Y} \text{ (em termos decimais)} \quad (56)$$

Ou ainda,

$$Y' = \arcsen\sqrt{\frac{Y}{100}} \text{ (em termos percentuais)} \quad (57)$$

Outra técnica de transformação chama-se boxcox, a qual pode ser aplicada a qualquer tipo de variável e resolve a grande maioria dos casos, principalmente para os casos nos quais as transformações anteriormente mencionadas não podem ser aplicadas. Seja uma variável aleatória mensurada  $Y$ , tem-se a seguinte transformação:

$$Y' = \begin{cases} \ln(Y) & \text{se } \lambda = 0 \\ \frac{Y^\lambda - 1}{\lambda} & \text{se } \lambda \neq 0 \end{cases} \quad (58)$$

O valor de  $\lambda$  deve ser encontrado, este valor maximiza o logaritmo da função de verossimilhança para uma distribuição normal. Alguns softwares disponibilizam *toolboxes* para a busca desse valor, utilizando a função boxcox presente neles.

## 2.4. Otimização Substituta

Considera-se um substituto como uma função que se assemelha a outra função. O substituto se torna útil porque leva menos tempo para ser avaliado. Quando se trata de funções objetivas demoradas, é indicada a utilização da otimização substituta.

A otimização substituta tenta encontrar um mínimo global de uma função usando poucas avaliações da mesma. O algoritmo, então, tenta equilibrar o processo de otimização entre dois objetivos: exploração e velocidade. O mesmo converge para uma solução global para funções objetivo contínuas em domínios limitados (GUTMANN, 2001).

Para Guo (2020), é necessário ter um modelo substituto para que seja alcançada a otimização substituta. Para isso, deve-se seguir algumas etapas, conforme descritas a seguir:

No MATLAB®, a função que realiza este tipo de otimização é chamada “surrogareopt”, conforme descrita por Mathworks (2019) e apresentada nesta seção. O algoritmo alterna entre duas fases: A construção de um substituto e a procura pelo mínimo da função objetivo.

Para construir o substituto, o algoritmo escolhe pontos quase aleatórios dentro dos limites e avalia a função objetivo nesses pontos. Feito isso, o algoritmo constrói um substituto a partir da interpolação da função objetivo por meio de um interpolador de base radial, esta interpolação possui algumas propriedades importantes que merecem destaque:

- O interpolador é definido através da mesma fórmula em qualquer número de dimensões e com qualquer número de pontos;
- Assume os valores prescritos nos pontos avaliados;
- Economiza tempo em sua avaliação e na adição de um ponto;
- Envolve a resolução de um sistema linear de equações  $n \times n$ , onde  $n$  corresponde ao número de pontos substitutos. Esse sistema possui solução única para muitos interpoladores (Powell, 1990).

A busca pelo mínimo da função objetivo segue um procedimento relacionado à pesquisa local. O solver inicia sua busca com uma escala inicial a

partir do ponto com o menor valor da função objetivo e procura um mínimo de uma função de mérito que se relacione tanto com o substituto quando com a distância dos pontos de busca existentes. A função de mérito  $f_{mer}(x)$  é uma combinação ponderada de dois termos, substituto em escala e distância em escala.

## 2.5. Modelos Substitutos

Modelos substitutos são modelos interpretáveis treinados para aproximarem as previsões de um modelo de caixa preta. São utilizados em engenharia como um modelo barato e rápido que capaz de substituir modelos mais robustos e demorados. A ideia de modelos substitutos pode ser encontrada com diferentes nomes: metamodelo, modelo de aproximação, modelo de superfície de resposta, entre outros (MOLNAR, 2021).

De acordo com Molnar (2021), deve-se seguir algumas etapas para a obtenção de um modelo substituto:

- 1) Seleção de um conjunto de dados;
- 2) Para o conjunto de dados selecionados, obtenha as previsões do modelo de caixa preta;
- 3) Seleção de um tipo de modelo interpretável
- 4) Treino do modelo interpretável no conjunto de dados;
- 5) Medição da eficiência do modelo obtido;
- 6) Interpretação do modelo substituto.

Uma forma de medir a eficiência do modelo substituto é analisando o  $R^2$ , a partir da seguinte equação:

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^n (\hat{y}_*^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^n (\hat{y}^{(i)} - \bar{\hat{y}})^2} \quad (59)$$

Onde  $\hat{y}_*^{(i)}$  é a previsão para a  $i$ -ésima instância do modelo substituto;  $\hat{y}^{(i)}$  corresponde à previsão do modelo de caixa preta;  $\bar{\hat{y}}$  corresponde à média das previsões do modelo de caixa preta; SSE representa a soma dos quadrados dos erros; SST representa a soma dos quadrados totais.

Para Simpson et al. (2001), considerando a análise de uma função dada por:

$$y = f(x) \quad (60)$$

Então um modelo substituto, ou um metamodelo, será dado por:

$$\hat{y} = g(x) \quad (61)$$

E então:

$$y = \hat{y} + \epsilon \quad (62)$$

Onde  $\epsilon$  representa tanto o erro obtido pela aproximação entre os modelos quanto o erro aleatório de medida. O metamodelo pode permitir ao usuário compreender um melhor entendimento de relação entre  $x$  e  $y$ , adquirir uma integração mais fácil e obter ferramentas de análise rápida para otimização e exploração do espaço de design por meio de aproximações.

### 2.5.1. Funções de Base Radial

Uma função de base radial é uma função utilizada para aproximar funções fornecidas e se mostra eficaz e flexível de forma que é bastante utilizada em aplicações de engenharia (BUHMANN, 2003).

De acordo com Fasshauer (2007), os tipos mais comuns de funções de bases radiais são as seguintes, onde  $r = ||x - x_i||$  e  $\epsilon$  indica um parâmetro de estimativa:

- Gaussiana

$$\varphi(r) = e^{-(\epsilon r)^2} \quad (63)$$

- Multiquadrática

$$\varphi(r) = \sqrt{1 + (\epsilon r)^2} \quad (64)$$

- Quadrática inversa

$$\varphi(r) = \frac{1}{1 + (\epsilon r)^2} \quad (65)$$

- Multiquadrática inversa



$$\varphi(r) = \frac{1}{\sqrt{1+(\varepsilon r)^2}} \quad (66)$$

- Spline poliarmônica

$$\begin{aligned} \varphi(r) &= r^k, & k=1,3,5\dots \\ \varphi(r) &= r^k \ln(r), & k=2,4,6\dots \end{aligned} \quad (67)$$

- Spline de placa fina

$$\varphi(r) = r^2 \ln(r) \quad (68)$$

As funções de base radial são geralmente usadas para contruir as funções de aproximação da forma como mostrada na equação a seguir, apresentada por Fasshauer (2007):

$$y(x) = \sum_{i=1}^N w_i \varphi(|x - x_i|) \quad (69)$$

Onde a função de aproximação  $y(x)$  é representada como o somatório de funções de base radial, cada uma associada a um centro diferente  $x_i$  e ponderada por um coeficiente apropriado  $w_i$ .

### 2.5.2. Modelos de Regressão Linear Múltipla

Na aplicação da análise de regressão, é comum se deparar com situações onde é necessário observar mais de uma variável independente, assim um modelo de regressão múltiplo se torna útil. Quando este modelo é linear, é chamado de Modelo de Regressão Linear Múltipla.

De acordo com Walpole et al. (2011), o modelo de regressão linear múltipla pode ser explicado a seguir. Considerando uma situação com  $k$  variáveis independentes  $x_1, x_2, \dots, x_k$ , a média de  $Y|x_1, x_2, \dots, x_k$  é dada pelo modelo mostrado na equação a seguir:

$$\mu_{Y|x_1, x_2, \dots, x_k} = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k \quad (70)$$

E a resposta estimada é obtida pela seguinte equação de regressão:

$$\hat{y} = b_0 + b_1 x_1 + \dots + b_k x_k \quad (71)$$

Onde cada coeficiente de regressão  $\beta_i$  é estimado por  $b_i$  a partir dos dados de amostra utilizando o método dos mínimos quadrados.

Conforme Walpole et al. (2011), as técnicas podem ser aplicadas quando o modelo linear envolver, por exemplo, potências e produtos de variáveis. Nestas situações pode-se aplicar um modelo de regressão polinomial, conforme mostrado na equação a seguir, considerando  $k = 1$ :

$$\mu_{Y|x} = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_r x^r \quad (72)$$

E a resposta estimada é obtida pela seguinte equação de regressão:

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \dots + b_r x^r \quad (73)$$

É possível obter os estimadores de mínimos quadrados  $\beta_0, \beta_1, \dots, \beta_k$  ajustando o modelo de regressão linear múltipla, mostrada na Equação 63 aos pontos de dados:

$$\{(x_{1i}, x_{2i}, \dots, x_{ki}, y_i); \quad i = 1, 2, \dots, n \text{ e } n > k\} \quad (74)$$

Onde  $y_i$  corresponde à resposta observada dos valores  $x_{1i}, x_{2i}, \dots, x_{ki}$  de  $k$  variáveis independentes  $x_1, x_2, \dots, x_k$ . Assume-se que cada observação  $x_{1i}, x_{2i}, \dots, x_{ki}, y_i$  satisfaz a seguinte equação:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \epsilon_i \quad (75)$$

Ou,

$$y_i = \hat{y}_i + e_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_k x_{ki} + e_i \quad (76)$$

Onde  $\epsilon_i$  e  $e_i$  correspondem, respectivamente, aos erros aleatório e residual associados com a resposta  $y_i$  e o valor ajustado  $\hat{y}_i$ .

No caso de uma regressão linear simples, assume-se que  $\epsilon_i$  é independente e identicamente distribuído com média 0 e variância comum  $\sigma^2$ . Para se chegar às estimativas pela minimização da expressão:

$$SSE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - b_0 - b_1 x_{1i} - b_2 x_{2i} - \dots - b_k x_{ki})^2 \quad (77)$$

### 2.5.3. Kriging

O método de Kriging foi criado por Krige (1951) e desenvolvida alguns anos depois por Matheron (1971). Este método utiliza uma relação chamada de semivariograma (relação espacial que os dados têm entre si; variação quadrática dada uma distância) e tendência (valor médio dos dados). Com estas duas

informações, o Kriging executa a interpolação por meio de uma média ponderada dos dados amostrais de forma que o erro esperado seja minimizado (FAZIO, 2013).

Através do Kriging, pode-se conhecer o erro associado à predição dos valores estimados. Este erro é analisado através da variância da estimativa (YAMAMOTO e CONDE, 1999).

De acordo com Fazio (2013), o Kriging divide o dado  $z$  em duas partes: tendência  $t$  e ruído  $s$ . Onde a primeira representa o valor esperado ou valor médio e a segunda representa uma parte imprevisível, cuja média é igual a zero. Como mostra a equação a seguir (CRESSIE, 1993):

$$z = s + t \quad (78)$$

O valor esperado do ruído é zero, assim, o valor esperado dos dados é igual à tendência.

$$E(z) = E(s) + E(t) = 0 + E(t) \quad (79)$$

$$E(z) = E(t) \quad (80)$$

Para realizar a interpolação, o Kriging faz uma soma ponderada dos dados amostrais:

$$z_j = \sum_i \omega_i z_i \quad (81)$$

Onde  $i$  representa os pontos amostrais,  $j$  representa os pontos com valores que precisam ser interpolados e  $\omega$  representa o peso de cada dado amostral. Assim:

$$E(z_j) = E(\sum_i \omega_i z_i) \quad (82)$$

Como o Kriging é um método não viciado, o valor interpolado esperado é igual ao valor real esperado:

$$E(z_j) = \sum_i \omega_i E(z_i) \quad (83)$$

Os pesos são calculados de forma que o valor esperado do erro quadrático da interpolação seja o menor possível. A equação do erro médio quadrático é dada por:

$$\sigma^2 = E\left(\sum_i \omega_i z_i - z_j\right)^2 \quad (84)$$

Um outro conceito utilizado no Kriging é o semivariograma  $\gamma$  que representa a metade da diferença quadrática esperada entre os valores de dois pontos  $a$  e  $b$  cuja distância entre si é  $h$ , dado pela equação:

$$\gamma(h) = \frac{E(z_a - z_b)^2}{2} \quad (85)$$

Na equação original do erro 77 é possível substituir a expressão  $z = t + s$ . Resultando em:

$$\begin{aligned} \sigma^2 &= E\left(\sum_i \omega_i (t_i + s_i) - (t_j + s_j)\right)^2 = E\left(\sum_i \omega_i t_i + \sum_i \omega_i s_i - t_j - s_j\right)^2 \\ \sigma^2 &= \left(E(\sum_i \omega_i t_i) + E(\sum_i \omega_i s_i) - E(t_j) - E(s_j)\right)^2 = \left(\sum_i \omega_i E(t_i) + E(\sum_i \omega_i s_i) - \right. \\ &\quad \left. - E(t_j) - E(s_j)\right)^2 \quad (86) \end{aligned}$$

A partir da Equação 73, pode-se afirmar que:

$$E(z_j) = E(t_j) \quad (87)$$

$$E(z_i) = E(t_i) \quad (88)$$

$$E(t_j) = \sum_i \omega_i E(t_i) \quad (89)$$

O que significa dizer que, o valor esperado da tendência real é igual à soma ponderada do valor esperado da tendência da amostra. Então:

$$\sigma^2 = \left(E(\sum_i \omega_i s_i) - E(s_j)\right)^2 = E(\sum_i \omega_i s_i - s_j) \quad (90)$$

Desta forma, se conclui que a tendência não influencia na função erro. A etapa seguinte consiste em minimizar a função erro, desde que a restrição da tendência seja respeitada. Para esta etapa pode-se utilizar o Multiplicador de Lagrange (MORDECAI, 2003).

A função de Lagrange é definida por:

$$\begin{aligned} \Lambda(x, \lambda) &= f(x) + \lambda g(x) \\ g(x) &= c \quad (91) \end{aligned}$$

Onde  $c$  representa uma constante,  $f(x)$  representa a função a ser minimizada,  $g(x)$  é a restrição e  $\lambda$  é uma variável extra chamada de multiplicador de Lagrange. A minimização respeitando a restrição é realizada igualando a derivada a zero:

$$\begin{aligned}\nabla(f(x) + \lambda g(x)) &= 0 \\ g(x) &= c\end{aligned}\tag{92}$$

Aplicando para as equações do Kriging:

$$\nabla\left(\sigma^2 + \lambda_i \left(E(t_j) - \sum_i \omega_i E(t_i)\right)\right) = 0\tag{93}$$

$$E(t_j) - \sum_i \omega_i E(t_i) = 0\tag{94}$$

Onde  $\lambda_i$  representa o Multiplicador de Lagrange,  $t$  representa a tendência,  $\sigma^2$  representa a função erro,  $\omega$  representa os pesos,  $i$  representa os pontos amostrais e  $j$  representa os pontos a serem interpolados.

Dependendo de como a tendência é assumida, o método de Kriging é abordado de forma diferente. Para uma tendência igual a zero ele é chamado de Kriging Simples, se a tendência é uma constante de valor desconhecido o método pode ser chamado de Kriging Ordinário e se a tendência for um polinômio o método é chamado de Kriging Universal (CRESSIE, 1993).

No software MATLAB® existe uma toolbox para aproximações via método kriging chamada DACE (*Design and Analysis of Computer Experiments*). O DACE utiliza o método do kriging ordinário, o qual assume uma média constante em todo o domínio. De acordo com Martin e Simpson (2003), a modelagem do kriging consiste em duas partes:

- Regressão linear dos dados (parte A);
- Ajuste sistemático do modelo (parte B).

Dado um conjunto de  $m$  dados com  $X = [x_1, x_2, \dots, x_m]^T$  representando o vetor de dados de entrada e  $Y = [y_1, y_2, \dots, y_m]^T$  representando o vetor resposta. De acordo com Sacks, Welch e Mitchell (1989), é adotado um modelo  $\hat{y}$  que

expressa a resposta determinística  $y(x)$  para os  $m$  dados de entrada, através de um modelo de regressão  $F$  e uma função randômica estocástica.

$$\hat{y}(x) = F(\beta, l, x) + z(x) \quad (95)$$

Onde  $\beta$  representa os parâmetros de regressão e o  $z(x)$  um processo randômico estacionário gaussiano com média nula e covariância dada pela equação a seguir:

$$E(z(x_1), z(x_2)) = \sigma^2 R(\theta, x_1, x_2) \quad (96)$$

Onde  $\sigma^2$  representa a variância, e  $R$  representa a função de correlação espacial com parâmetros  $\theta$  a qual é responsável pelo controle da suavidade do modelo, da influência dos pontos adjacentes e da diferença na superfície de resposta.

Sejam pontos do modelo computacional:

$$X = \{x_1, x_2, \dots, x_n\} \subset \Omega \quad (97)$$

Onde  $\Omega$  representa todas as possíveis entradas do modelo que resultam em respostas, ou o domínio do modelo computacional. As respostas são dadas por:

$$Y = \{y(x_1), y(x_2), \dots, y(x_n)\} \quad (98)$$

Considera-se uma estimativa linear para as respostas:

$$\hat{y}(x) = \lambda^T(x)Y \quad (99)$$

Em qualquer ponto  $x \in \Omega$ . O kriging assume  $\hat{y}(x)$  como uma variável randômica e encontra a melhor estimativa linear imparcial,  $\lambda^T(x)Y$ , que minimiza o erro quadrado médio  $MSE$  da estimativa:

$$MSE[\hat{y}(x)] = E[\lambda^T(x)Y - y(x)]^2 \quad (100)$$

Sujeito a uma restrição de imparcialidade:

$$E[\lambda^T(x)Y] = E[y(x)] \quad (101)$$

O kriging universal é definido como um conjunto de funções de regressão:

$$f(x) = \{f_1(x), f_2(x), \dots, f_k(x)\}^T \quad (102)$$

O kriging ordinário é um caso especial, definido como:

$$f(x) = \{1\} \quad (103)$$

A próxima etapa é definir um vetor  $F$  que seja o valor de  $f(x)$  avaliado em cada um dos locais conhecidos:

$$F = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \quad (104)$$

Será necessário construir uma matriz de correlação. A matriz  $R$  representa a matriz de correlação composta pela função de correlação espacial avaliada e cada combinação possível de pontos conhecidos:

$$R = \begin{bmatrix} R(\theta, x_1, x_1) & R(\theta, x_1, x_2) & \dots & R(\theta, x_1, x_n) \\ R(\theta, x_2, x_1) & R(\theta, x_2, x_2) & \dots & R(\theta, x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ R(\theta, x_n, x_1) & R(\theta, x_n, x_2) & \dots & R(\theta, x_n, x_n) \end{bmatrix} \quad (105)$$

Por último é preciso definir um vetor que representa a correlação entre um ponto desconhecido e todos os pontos conhecidos:

$$r(x) = \{R(\theta, x, x_1), R(\theta, x, x_2), \dots, R(\theta, x, x_n)\}^T \quad (106)$$

Se  $\lambda(x)$  resolve o problema de minimização da restrição, então  $\lambda^T(x)Y$  é chamado de melhor estimativa linear imparcial para  $\hat{y}(x)$ , a qual pode ser encontrada a partir da equação:

$$\hat{y}(x) = f^T(x)\hat{\beta} + r^T(x)R^{-1}(Y - F\hat{\beta}) \quad (107)$$

Onde  $\hat{\beta}$  é conhecido como a estimativa dos mínimos quadrados generalizados.

$$\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} Y \quad (108)$$

O erro quadrado médio  $MSE$  da estimativa de  $\hat{y}(x)$  é dado pela equação seguinte:

$$MSE[\hat{y}(x)] = \sigma^2 (1 - [f^T(x) \quad r^T(x)] \begin{bmatrix} 0 & F^T \\ F & R \end{bmatrix}^{-1} \begin{bmatrix} f(x) \\ r(x) \end{bmatrix}) \quad (109)$$

A primeira parte da Equação (100) representa a estimativa do mínimo quadrado generalizado de um ponto  $x \in \Omega$  dada a correlação da matriz  $R$ . A segunda parte arrasta a superfície de resposta pelos pontos de dados conhecidos. A elasticidade da superfície de resposta é determinada pela função de correlação espacial  $R$ . As estimativas dos pontos são retornadas exatamente iguais às observações correspondentes usadas para criar a função, realizando a interpolação dos dados conhecidos. Neste ponto, o erro quadrado médio é igual a zero porque não há incerteza nos resultados do modelo. Como a estimativa de  $x$  se afasta dos pontos conhecidos, a segunda parte da equação (100) se aproxima de zero, obtendo a estimativa dos mínimos quadrados generalizados.

A toolbox fornece modelos de regressão polinomial de ordem 0, 1 e 2. Para a correlação dos dados, existem diferentes formas de funções como mostrado por Lophaven, Nielsen e Sondegaard (2002): Exponencial, gaussiana, linear, esférica, cúbica, spline, etc.

## **2.6. Construção de modelos substitutos disponíveis no MATLAB®**

No software MATLAB® existe a função “fitrgp”, que é responsável por fazer a construção do modelo substituto. De acordo com Mathworks (2019), esta função aceita qualquer combinação de métodos de ajuste, predição e seleção de conjunto ativo.

O ajuste do modelo consiste em estimar os seguintes parâmetros a partir dos dados: A função de covariância parametrizada  $k(x_i, x_j | \theta)$  em termos dos parâmetros de kernel no vetor  $\theta$ , o qual será mostrado posteriormente; A variância de ruído  $\sigma^2$ ; O vetor de coeficientes de funções de base fixa  $\beta$ .

Para Rasmussen (2006), em processos gaussianos a função de covariância expressa a similaridade entre os valores preditores  $x_i$  e as variáveis respostas  $y_i$ . Ela determina como a resposta no ponto  $x_i$  é afetada pelas respostas de outros pontos  $x_j$ . A função de covariância  $k(x_i, x_j)$  pode ser definida por várias funções de kernel e pode ser parametrizada em termos dos parâmetros de kernel no vetor  $\theta$ . Assim, é possível expressar a função de covariância como  $k(x_i, x_j | \theta)$ .



Segundo Rasmussen (2006), os parâmetros de kernel são baseados nos sinais de desvio padrão  $\sigma_f$  e na escala de comprimento característico  $\sigma_l$ . Esses dois parâmetros devem ser maiores do que zero, de modo que:

$$\theta_1 = \log \sigma_l \text{ e } \theta_2 = \log \sigma_f \quad (110)$$

As funções de covariância de kernel podem variar e algumas destas variações são mostradas a seguir.

- *Squared exponential*

$$k(x_i, x_j | \theta) = \sigma_f^2 \exp \left[ -\frac{1}{2} \frac{(x_i - x_j)^T (x_i - x_j)}{\sigma_l^2} \right] \quad (111)$$

- *Exponential*

$$k(x_i, x_j | \theta) = \sigma_f^2 \exp \left( -\frac{r}{\sigma_l} \right) \quad (112)$$

Onde  $r$  corresponde à distância Euclidiana entre  $x_i$  e  $x_j$ , dada pela seguinte equação:

$$r = \sqrt{(x_i - x_j)^T (x_i - x_j)} \quad (113)$$

- *Matern 3/2*

$$k(x_i, x_j | \theta) = \sigma_f^2 \left( 1 + \frac{\sqrt{3} r}{\sigma_l} \right) \exp \left( -\frac{\sqrt{3} r}{\sigma_l} \right) \quad (114)$$

Onde  $r$  corresponde à distância Euclidiana entre  $x_i$  e  $x_j$ , apresentada na Equação (113).

- *Matern 5/2*

$$k(x_i, x_j) = \sigma_f^2 \left( 1 + \frac{\sqrt{5} r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2} \right) \exp \left( -\frac{\sqrt{5} r}{\sigma_l} \right) \quad (115)$$

Onde  $r$  corresponde à distância Euclidiana entre  $x_i$  e  $x_j$ , apresentada na Equação (113).

- *Rational Quadratic*

$$k(x_i, x_j | \theta) = \sigma_f^2 \left( 1 + \frac{r^2}{2\alpha\sigma_l^2} \right)^{-\alpha} \quad (116)$$

Onde  $\alpha$  corresponde a um parâmetro de mistura de escala com valor positivo e  $r$  corresponde à distância Euclidiana entre  $x_i$  e  $x_j$ , apresentada na Equação (113).

- *ARD Squared Exponential*

$$k(x_i, x_j | \theta) = \sigma_f^2 \exp \left[ -\frac{1}{2} \sum_{m=1}^d \frac{(x_{im} - x_{jm})^2}{\sigma_m^2} \right] \quad (117)$$

- *ARD Exponential*

$$k(x_i, x_j | \theta) = \sigma_f^2 \exp(-r_2) \quad (118)$$

Onde  $r_2$  corresponde à seguinte equação:

$$r_2 = \sqrt{\sum_{m=1}^d \frac{(x_{im} - x_{jm})^2}{\sigma_m^2}} \quad (119)$$

- *ARD Matern 3/2*

$$k(x_i, x_j | \theta) = \sigma_f^2 (1 + \sqrt{3} r_2) \exp(-\sqrt{3} r_2) \quad (120)$$

Onde  $r_2$  corresponde à Equação (119).

- *ARD Matern 5/2*

$$k(x_i, x_j | \theta) = \sigma_f^2 \left( 1 + \sqrt{5} r_2 + \frac{5}{3} r_2^2 \right) \exp(-\sqrt{5} r_2) \quad (121)$$

Onde  $r_2$  corresponde à Equação (119).

- *ARD Rational Quadratic*

$$k(x_i, x_j | \theta) = \sigma_f^2 \left( 1 + \frac{1}{2\alpha} \sum_{m=1}^d \frac{(x_{im} - x_{jm})^2}{\sigma_m^2} \right)^{-\alpha} \quad (122)$$

- *Matern 1/2*

$$k(x_i, x_j | \theta) = \sigma_f^2 \exp\left(-\frac{r}{\sigma_1}\right) \quad (123)$$

Onde  $r$  corresponde à distância Euclidiana entre  $x_i$  e  $x_j$ , apresentada na Equação (113).

- *ARD Matern 1/2*

$$k(x_i, x_j | \theta) = \sigma_f^2 \exp(-r) \quad (124)$$

## 2.7. Medidas de um bom modelo

### 2.7.1. Correlação máxima de pares

Os testes de significância descritos são adequados para determinar quais variáveis devem ser usadas no modelo de regressão final. De acordo com Walpole et al. (2011), os testes  $t$  podem ser úteis em muitos problemas onde o número de variáveis investigadas é pequeno. Entretanto, quando o experimento exibe um desvio da ortogonalidade, é necessário usar técnicas mais elaboradas.

Medidas úteis de dependência linear entre as variáveis independentes são fornecidos pelos coeficientes de correlação da amostra  $r_{ij}$ . Quando um ou mais coeficientes desvia consideravelmente do valor zero, o processo de encontrar o subconjunto mais eficaz pode se tornar muito difícil (HOCKING, 1976).

De acordo com Walpole et al. (2011) o usuário de regressão linear múltipla tende a alcançar um dos objetivos a seguir: Obter estimativas de coeficientes individuais em um modelo completo; Observar as variáveis com efeitos significativos na resposta; Chegar à equação de previsão mais eficaz. Para cada um desses objetivos, a multicolinearidade no experimento pode ter efeito na regressão.

A seguir, é apresentado um procedimento padrão para pesquisar o subconjunto ideal de variáveis, apresentado por Walpole et al. (2011). A seleção é baseada no procedimento de que as variáveis devem ser inseridas uma a uma até que a equação de regressão seja satisfatória.

1. Escolher a variável que dá o maior  $R^2$  e chamá-la de  $x_1$ . Se  $x_1$  for insignificante, o procedimento é encerrado.
2. Escolher a variável que cause aumento em  $R^2$ , na presença de  $x_1$ . Esta variável será chamada de  $x_j$ , para o qual:

$$R(\beta_j | \beta_1) = R(\beta_1, \beta_j) - R(\beta_1) \quad (125)$$

Seja suficientemente grande. Na Equação (116), o termo  $\beta$  corresponde aos coeficientes individuais na equação da regressão. A variável  $x_j$  passa a ser chamada de  $x_2$ . O modelo de regressão é então ajustado e o parâmetro  $R^2$  é observado. Se  $x_2$  for insignificante, o procedimento é encerrado.

3. Escolher a variável  $x_j$  que dá o maior valor de:

$$R(\beta_j | \beta_1, \beta_2) = R(\beta_1, \beta_2, \beta_j) - R(\beta_1, \beta_2) \quad (126)$$

Novamente resultando no maior aumento de  $R^2$  em relação à etapa anterior. Chama-se essa variável de  $x_3$ . Se  $x_3$  for insignificante, procedimento é encerrado.

Este processo continua até que a última variável inserida não cause um aumento significativo no valor de  $R^2$ . Esse aumento pode ser determinado em cada etapa através do *teste f* ou do *teste t*. Por exemplo, na etapa 2, a equação a seguir pode ser utilizada para testar a variável  $x_2$  no modelo:

$$f = \frac{R(\beta_2|\beta_1)}{s^2} \quad (127)$$

Onde  $s^2$  representa o erro quadrático médio que contém as variáveis  $x_1$  e  $x_2$ . De forma semelhante, na etapa 3, a variável  $x_3$  pode ser testada através da equação a seguir:

$$f = \frac{R(\beta_3|\beta_1,\beta_2)}{s^2} \quad (128)$$

Onde  $s^2$  representa o erro quadrático médio que contém as variáveis  $x_1$ ,  $x_2$  e  $x_3$ .

Se  $f < f_\alpha(1, n - 3)$  na etapa 2 para um determinado nível de significância,  $x_2$  não é incluído e o processo é finalizado, resultando em uma equação linear simples que relaciona  $y$  e  $x_1$ . Entretanto, se  $f > f_\alpha(1, n - 3)$ , deve-se seguir para a etapa 3. Na etapa 3, se  $f < f_\alpha(1, n - 4)$ ,  $x_3$  não é incluído e o processo é finalizado, resultando em uma equação que contém as variáveis  $x_1$  e  $x_2$ .

### 2.7.2. Discrepância L2 modificada

Em um modelo de regressão linear, um projeto ortogonal é desejável pois fornece estimativas não correlacionadas dos coeficientes e melhora o desempenho da classificação e dos modelos de regressão (KIM e LOH, 2003). Um bom preenchimento é aquele que os pontos estão espalhados por toda região experimental. A correlação  $\rho$  é o critério mais usado para avaliar o Latin Hipercubo. Hernandez, Lucas e Carlyle (2012) apresentam o cálculo do coeficiente de correlação entre quaisquer duas colunas vetoriais,  $X_i$  e  $X_j$ , na matriz de projeto  $X$ , conforme mostrado na equação a seguir.

$$\rho_{ij} = \frac{\sum_{b=1}^n [(X_b^i - \bar{X}^i)(X_b^j - \bar{X}^j)]}{\sqrt{\sum_{b=1}^n (X_b^i - \bar{X}^i)^2 \sum_{b=1}^n (X_b^j - \bar{X}^j)^2}} \quad (129)$$

Onde  $\bar{X}^i$  e  $\bar{X}^j$  representam as médias dos valores da  $i$ -ésima e da  $j$ -ésima coluna na matriz  $X$ .

Em um projeto experimental  $n \times k$ , existe  $\binom{k}{2}$  correlação de pares, o maior em magnitude define o grau de não ortogonalidade no projeto. A equação a seguir expressa a correlação absoluta máxima de pares:

$$\rho_{map} = \max\{|\rho_{ij}|, \forall (i \neq j)\} \quad (130)$$

Ao minimizar o  $\rho_{map}$  limitamos a correlação de pares de pior caso. Para um projeto envolvendo muitos fatores, uma baixa correlação média absoluta não garante que todas as correlações de pares sejam pequenas.

Uma das escolhas de medida de preenchimento de espaço é a discrepância L2 modificada com eficiência computacional. Hernandez, Lucas e Carlyle (2012) apresentaram uma equação que usa a discrepância para medir o preenchimento do espaço, mostrada a seguir:

$$ML_2 = \binom{4}{3}^k - \frac{2^{1-k}}{n} \sum_{d=1}^n \prod_{i=1}^k (3 - x_{di}^2) + \frac{1}{n^2} \sum_{d=1}^n \sum_{j=1}^n \prod_{i=1}^k [2 - \max(x_{di}, x_{ji})] \quad (131)$$

Se o projeto experimental tem um baixo  $ML_2$ , significa que todas as projeções do projeto em subconjuntos de  $k$  variáveis também terão boa uniformidade.

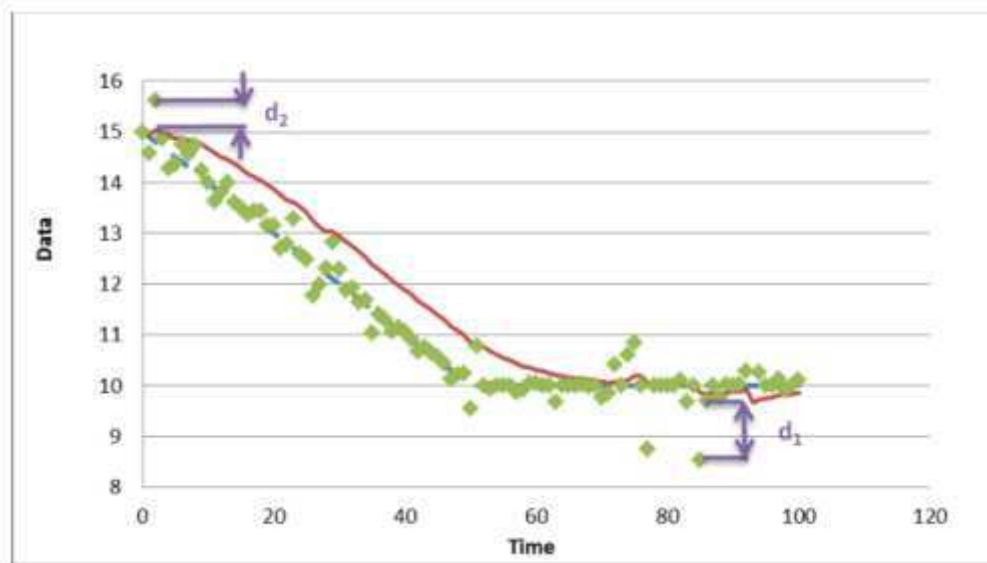
## 2.8. Detecção do estado estacionário

A identificação do estado estacionário em processos que possuam algum tipo de ruído é extremamente importante e amplamente utilizada em controles de processo e otimização.

Rhinehart (2013) apresenta um método computacional de simples execução para essa detecção e será apresentado nesta seção. As variáveis envolvidas em um processo químico quase sempre apresentam ruídos. Então, o método de detecção do estado estacionário precisa analisar os dados através desses ruídos e determinar quando ocorrerá o estado estacionário. Para ser eficiente, não é interessante que se analise apenas as amostras mais recentes, ele precisa analisar uma tendência local.

O motivo dos ruídos atribui-se a: consequência de autocorrelação de tendências, amplitude de ruído variável, picos individuais, distribuições de ruídos não gaussianas ou eventos adversos.

O método proposto por Rhinehart (2013) utiliza estatística R e uma razão de duas variâncias. A Figura 2 apresenta os dados de um processo.



**Figura 2: Dados de um processo, com ruído e desvios.**  
**Fonte: Rhinehart (2013)**

A linha pontilhada (azul) representa a verdadeira tendência do processo. Os pontos (verdes) representam os dados medidos e é possível observar pequenas variações (ruídos).

O método começa calculando uma tendência filtrada das medidas do processo (linha vermelha). A variância é medida por dois métodos:  $d_2$  representa a diferença entre a medição e a tendência filtrada;  $d_1$  representa a diferença entre as medidas.

Se o processo estiver em estado estacionário, o valor filtrado da medida ( $X_f$ ) estará quase no meio dos valores do processo (Tempo de 80 a 100 no gráfico). Então, a variância do processo,  $\sigma^2$ , estimada por  $d_2$  será igual a  $\sigma^2$  estimada por  $d_1$ . Isso significa que a razão entre as variâncias será aproximadamente 1, como apresentado na equação a seguir.

$$r = \frac{\sigma^2 d_2}{\sigma^2 d_1} \cong 1 \quad (132)$$

O valor filtrado é determinado pela seguinte equação:

$$X_{f,i} = \lambda_1 X_i + (1 - \lambda_1) X_{f,i-1} \quad (133)$$

Onde  $X_i$  representa a variável do processo,  $X_f$  representa o valor filtrado de  $X$  e  $\lambda_1$  representa o fator do filtro.

A equação a seguir apresenta um método de obtenção da variância.

$$v_{f,i}^2 = \lambda_2(X_i - X_{f,i-1})^2 + (1 - \lambda_2)v_{f,i-1}^2 \quad (134)$$

Onde  $v_f^2$  representa a medida da variância do valor filtrado baseada na diferença entre dados e os valores filtrados.

O segundo método para obter a medida da variância é representado pela seguinte equação:

$$\delta_{f,i}^2 = \lambda_3(X_i - X_{f,i-1})^2 + (1 - \lambda_3)\delta_{f,i-1}^2 \quad (135)$$

Assim, a razão entre as medidas das variâncias, determinada de estatística-R, pode ser obtida pela equação a seguir.

$$R = \frac{(2-\lambda_1)v_{f,i}^2}{\delta_{f,i}^2} \quad (136)$$

## 2.9. Caixa de ferramentas GPML

A caixa de ferramentas GPML (*Gaussian Processes Machine Learning*) fornece uma gama de opções para inferência de processos gaussianos e previsão. Os processos gaussianos são especificados por funções de média e covariância (RASMUSSEN e NICKISCH, 2010).

A caixa de ferramentas está disponível de forma gratuita na web e é de fácil uso. Uma distribuição de processo gaussiano em uma função latente desconhecida é dada por:

$$f \sim \mathcal{GP}(m_\phi(x), k_\psi(x, x')) \quad (137)$$

E consiste em uma função de média:

$$m(x) = \mathbb{E}[f(x)] \quad (138)$$

E uma função de covariância:

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x')))] \quad (139)$$

Ambos contém hiperparâmetros  $\phi$  e  $\psi$  que se tem interesse de encaixar os dados. Geralmente assume-se observações independentes de entrada/saída  $(x_i, y_i)$  de  $f$  com probabilidade conjunta dada por:

$$\mathbb{P}_\rho(y|f) = \prod_{i=1}^n \mathbb{P}_\rho(y_i|f(x_i)) \quad (140)$$

Após especificação e ajuste dos hiperparâmetros  $\theta = \{\phi, \psi, \rho\}$ , deseja-se calcular distribuições preditivas para casos teste.

Rasmussen e Nickisch (2018) apresentaram o manual completo da ferramenta. Aqui será apresentada a síntese das funções de covariância utilizadas nesse trabalho.

Uma função de covariância  $k_\psi: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  (com hiperparâmetros  $\psi$ ) de um processo gaussiano  $f$  é uma função escalar definida sobre todo o domínio  $\mathcal{X}^2$  que calcula a covariância dada pela Equação 139, de  $f$  entre os dados de entrada  $x$  e  $z$ .

$$k(x, z) = \mathbb{V}[f(x), F(z)] = \mathbb{E}[f(x) - m(x))(f(z) - m(z))] \quad (141)$$

Para a resolução do modelo, é necessário apenas a avaliação da matriz de covariância completa dada por:

$$K = k_\psi(X) \quad (142)$$

E seus derivados:

$$K_i = \frac{\partial}{\partial \psi_i} K \quad (143)$$

Bem como os termos cruzados:

$$k_* = k_\psi(X, x_*) \text{ e } k_{**} = k_\psi(x_*, x_*) \quad (144)$$

A ferramenta oferece uma longa lista de funções de covariância simples e compostas.

Um hiperprévio  $p(\theta)$  com  $\theta = [\rho, \phi, \psi]$  é uma distribuição de probabilidade conjunta sobre os hiperparâmetros de probabilidade  $\rho$ , os hiperparâmetros médios  $\phi$  e os hiperparâmetros de covariância  $\psi$ . Estes hiperprévios podem ser usados para regularizar a otimização dos hiperparâmetros através da



probabilidade marginal  $Z(\theta)$  de tal forma que  $p(\theta)Z(\theta)$  é maximizado. A ferramenta também oferece uma longa lista de funções para os hiperparâmetros.

Neste trabalho, as funções de covariância utilizadas são as mesmas da função “surrogateopt” e já foram apresentadas na seção 2.6.

## 2.10. Validação Cruzada

A validação cruzada consiste em uma técnica para avaliar um modelo em um novo conjunto de dados. A técnica utilizada nesta tese consiste em dividir o conjunto de dados em grupos (folds) e então utiliza alguns grupos para estimação dos parâmetros do modelo e outros grupos para validação (KOHAVI, 1995) .

A precisão do modelo é dada pela seguinte equação:

$$Ac = \frac{1}{v} \sum_{i=1}^v (y_i - \hat{y}_i) \quad (145)$$

Onde  $y_i$  representa o valor da variável resposta real e  $\hat{y}_i$  representa o valor da variável resposta prevista pelo modelo.

Existem alguns diferentes tipos de validação cruzada e serão apresentados resumidamente nesta seção:

### 2.10.1. K-fold

Este tipo de validação é a utilizada neste trabalho. Nesta técnica, a amostra original é dividida em “k” grupos de tamanhos iguais. Entre esses grupos, um será reservado para testar o modelo e os grupos restantes serão usados como dados de treinamento. O processo é repetido “k” vezes, onde cada grupo será usado uma vez como dado de validação. O valor de “k” mais utilizado e indicado na literatura é igual a 10 (McLACHLAN, 2004).

Por exemplo, se  $k = 2$ , o conjunto será dividido em dois grupos de amostras com tamanhos iguais. Então, primeiramente o algoritmo irá treinar o primeiro grupo e validar no segundo grupo, e depois, treinar no segundo grupo e validar no primeiro grupo.

### **2.10.2. Holdout**

No método holdout, são atribuídos pontos arbitrários a dois conjuntos, sendo um conjunto de treinamento e um conjunto de testes. Na validação cruzada convencional, os resultados de vários testes são calculados juntos. Já no método *holdout*, tudo é feito em uma única execução, o que pode levar a resultados enganosos. Essa é considerada uma validação extremamente simples (KOHAVI, 1995).

### **2.10.3. Validação de subamostragem aleatória repetida**

Este método também é conhecido como validação cruzada de Monte Carlo. Ele divide o conjunto de dados aleatoriamente em dados de treinamento e dados de validação. Em cada divisão, o modelo é ajustado aos dados de treinamento e avaliado nos dados de validação. A desvantagem deste método é que algumas amostras podem não ser selecionadas para validação, enquanto outras podem ser selecionadas mais de uma vez (KUHN e JOHNSON, 2013).

### **2.10.4. Leave-p-out**

Esta técnica envolve o uso de “p” amostras como sendo o conjunto de validação e as demais amostras como sendo o conjunto de treinamento. Isto é repetido em todas as maneiras capazes de separar “p” novas amostras para serem tratadas como validação (CELISSE 2014).

Aqui é necessário que o treinamento e validação do modelo seja feito  $C_p^n$  vezes. Onde  $n$  é o número de amostras do conjunto original e  $C_p^n$  é o coeficiente binomial. Uma variante onde usa  $p = 2$  tem sido recomendada como um método imparcial (AIROLA et al., 2011).

### **2.10.5. Leave-one-out**

Trata-se de uma técnica particular do caso “leave-p-out”, quando  $p = 1$ . Este tipo de validação exige menos esforço computacional porque a combinação binomial será  $C_1^n$ . Ainda assim, caso  $n$  seja relativamente grande, um esforço computacional alto é necessário. Por este motivo a validação cruzada do tipo “k-fold” representa uma das melhores técnicas de validação cruzada (MOLINARO, SIMON e PFEIFFER, 2005).

### 2.11. Método dos mínimos quadrados (OLS) com seleção de variáveis

Para Guyon e Elisseeff (2003), o recurso de seleção de variáveis é indicado quando se deseja encontrar modelos de sistemas com inúmeras variáveis e este recurso tem o objetivo de: Melhorar o desempenho de previsão do modelo; fornecer regressores mais rápidos e econômicos; fornecer uma melhor compreensão do processo gerador de dados.

Os algoritmos de seleção de variáveis procuram um subconjunto de regressores que sejam capazes de modelar as respostas de maneira ideal. Entretanto, o uso de muitos recursos pode prejudicar o desempenho da previsão, mesmo quando todos os recursos são relevantes e contém informações sobre a variável resposta.

Existem três tipos de algoritmos de seleção de variáveis e serão apresentados a seguir:

- **Filtro:** mede a importância da variável com base nas características das mesmas, por exemplo, a variância. São selecionadas variáveis importantes como etapa de pré-processamento de dados e, em seguida, treina um modelo usando as variáveis selecionadas;
- **Envoltório:** Neste tipo, o algoritmo inicia o treinamento usando um subconjunto de variáveis e, adiciona ou remove uma variável baseado em um critério de seleção, observando o desempenho do algoritmo. O algoritmo, então, repete o treinamento e a melhoria de um modelo até que seus critérios de parada sejam satisfeitos;
- **Incorporado:** Esta técnica consegue identificar a importância das variáveis selecionadas após o treinamento do modelo. Ele é capaz de selecionar recursos que funcionam bem com um determinado processo de aprendizagem.

No Matlab há diversas funções para tipo de método de seleção de variáveis, como por exemplo: “fscchi2”, “fscmr”, “fitlinear”, “fitgpr”, “fitrlinear”, “sequentialfs”.

Neste trabalho será utilizada a função “sequentialfs”, a qual corresponde ao tipo envoltório. Ela é indicada para ser usada em problemas de classificação e regressão.

A função “sequentialfs” possui as seguintes características (MATHWORKS, 2019):

- 1) Seleção de variáveis sequenciais baseado em um critério definido;
- 2) Definição de uma função que implemente um algoritmo de aprendizado;
- 3) Especificação da seleção sequencial “para frente”, “para trás” ou ambos;
- 4) Utilização de validação cruzada para avaliar o critério.

A função aplicada é descrita em detalhes na metodologia deste trabalho. Basicamente, a função seleciona um subconjunto de variáveis da matriz de dados X que melhor prevê os dados em y selecionando variáveis de forma sequencial até que não haja melhoria na previsão do modelo. Para cada subconjunto de variáveis candidatas, a função “sequentialfs” realiza validação cruzada do tipo “k-fold” com o número de “folds” igual a 10.

## **2.12. Otimização Bayesiana**

A essência da otimização bayesiana é a atualização de um entendimento anterior para produzir novas informações. A otimização bayesiana é indicada para casos de otimizações em que as funções apresentam as seguintes características: Exige um grande esforço computacional; A derivada é desconhecida; É necessário encontrar os mínimos globais da função (YE, 2020).

Segundo Mathworks (2019), a otimização bayesiana mantém um modelo de processo gaussiano da função objetivo internamente e usa avaliações da função para treinar o modelo, além de utilizar uma função de aquisição para

determinar o próximo ponto a ser avaliado. Essa função de aquisição pode equilibrar a amostragem e explorar áreas que ainda não foram modeladas.

Esta otimização é adequada para otimizar hiperparâmetros (parâmetros interno) de algoritmos de classificação e regressão. Esses parâmetros podem afetar o desempenho da modelagem e normalmente são difíceis de serem otimizados. Otimizar hiperparâmetros significa tentar minimizar a perda de validação cruzada do modelo.

Para Jones, Schonlau e Welch (1998), existem muitas funções de aquisição mas a função de melhoria esperada, mostrada na Equação 146 é uma escolha comum, pois é possível ser calculada de forma fechada se a previsão do modelo  $y$  seguir uma distribuição normal mostrada na Equação 147.

$$\mathbb{E}[\mathbb{I}(\lambda)] = \mathbb{E}[\max(f_{min} - y, 0)] \quad (146)$$

$$\mathbb{E}[\mathbb{I}(\lambda)] = (f_{min} - \mu(\lambda))\Phi\left(\frac{f_{min} - \mu(\lambda)}{\sigma}\right) + \sigma\phi\left(\frac{f_{min} - \mu(\lambda)}{\sigma}\right) \quad (147)$$

Onde  $\phi(\cdot)$  e  $\Phi(\cdot)$  são a densidade normal padrão e a função de distribuição normal padrão e  $f_{min}$  é o melhor valor observado até agora.

A otimização gaussiana emprega processos gaussianos para modelar a função objetivo. Um processo gaussiano é completamente especificado por uma média e uma função de covariância. Porém, na otimização bayesiana, essa média é geralmente assumida como sendo uma constante.

Previsões de média  $\mu(\cdot)$  e variância  $\sigma^2(\cdot)$  podem ser obtidas pelas equações a seguir:

$$\mu(\lambda) = k_*^T K^{-1} y \quad (148)$$

$$\sigma^2(\lambda) = k(\lambda, \lambda) - k_*^T K^{-1} k_* \quad (149)$$

Onde  $k_*$  representa o vetor de covariância entre  $\lambda$  e todas as observações anteriores;  $K$  representa a matriz de covariância de todas as configurações avaliadas anteriormente e  $y$  são os valores observados das funções (SNOEK, LAROCHELLE e ADAMS, 2012).

Conforme mencionado anteriormente, a função de aquisição é utilizada para selecionar o próximo ponto o qual a função será avaliada e são várias que podem ser utilizadas.

Kushner (1964) sugeriu maximizar a probabilidade de melhoria  $f(x^+)$ , onde  $x^+ = \operatorname{argmax}_{x_i \in x_{1:t}} f(x_i)$ . Assim:

$$PI(x) = P(f(x) \geq f(x^+)) = \Phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right) \quad (150)$$

Onde  $\Phi(\cdot)$  representa a função de distribuição cumulativa normal.

Uma função de aquisição mais satisfatória seria uma que leva em conta não apenas a probabilidade de melhoria, mas a magnitude da melhoria que um ponto pode produzir. Jones et al., (1998) propõe a seguinte função de aquisição:

$$EI(x) = \begin{cases} (\mu(x) - f(x^+))\Phi(Z) + \sigma(x)\phi(Z) & \text{se } \sigma(x) > 0 \\ 0 & \text{se } \sigma(x) = 0 \end{cases} \quad (151)$$

Onde:

$$Z = \frac{\mu(x) - f(x^+)}{\sigma(x)} \quad (152)$$

Uma outra função de aquisição é do tipo exploração/, a equação a seguir foi apresentada por Lizotte (2008).

$$EI(x) = \begin{cases} (\mu(x) - f(x^+) - \xi)\Phi(Z) + \sigma(x)\phi(Z) & \text{se } \sigma(x) > 0 \\ 0 & \text{se } \sigma(x) = 0 \end{cases} \quad (153)$$

Onde:

$$Z = \begin{cases} \frac{\mu(x) - f(x^+) - \xi}{\sigma(x)} & \text{se } \sigma(x) > 0 \\ 0 & \text{se } \sigma(x) = 0 \end{cases} \quad (154)$$

A variável  $\xi$  representa um ruído, um bom valor indicado é  $\xi = 0,01$ .

### 2.13. Técnicas de interpolação

Para escolher um método de interpolação, deve-se considerar as características dos dados e do método escolhido. Os métodos mais simples são: Vizinho anterior e interpolações lineares (DAN, DINSOREANU e MURESAN, 2020).

No método chamado “vizinho anterior”, é identificado o valor de dados na iteração anterior o intervalo de valores desconhecidos. Assim, os valores interpolados são inteiramente determinados pelo valor anterior conhecido (MATHWORKS, 2020).

Na interpolação linear, os novos pontos são estimados juntando com uma linha reta os valores conhecidos mais próximos localizados a esquerda e a direita por meio de polinômios lineares (SIAUW e BAYEN, 2014).

Existem alguns interpoladores mais complexos e completos. Por exemplo, o interpolador “Cubic Spline”. Este método estima novos pontos de dados através da junção de valores conhecidos por um polinômio cúbico, baseado em valores dos pontos vizinhos. Tem ainda o método de interpolação Akima que é baseado, também, em polinômios cúbicos e utiliza apenas os valores próximos vizinhos para estimar os novos ponto de dados (MATHWORKS, 2020).

A diferença entre esses métodos está no grau dos polinômios. Entre os interpoladores de baixa complexidade, a interpolação linear utiliza polinômio de primeiro grau e o método de vizinho anterior utiliza um polinômio de grau zero também conhecido como função constante. Os interpoladores mais complexos apresentados aqui, “Cubic Spline” e Akima, utilizam polinômio de terceiro grau.

Neste trabalho será utilizada a interpolação Akima para identificar o  $Q^2$  médio após detecção do estado estacionário. Portanto, o método de interpolação Akima será apresentado detalhadamente.

### **2.13.1. Método de interpolação Akima**

Akima (1970) desenvolveu um método de interpolação para um conjunto de dados em um plano e para gerar uma curva suave entre os pontos. É baseado em uma função composta por polinômios, cada um com grau três.

Assume-se que a inclinação da curva em cada ponto é determinada localmente pelas coordenadas de cinco pontos. Sejam os pontos 1, 2, 3, 4, 5, a inclinação  $t$  da curva no ponto 3 é dada pela seguinte equação.

$$t = \frac{(|m_4 - m_3|)m_2 + (|m_2 - m_1|)m_3}{(|m_4 - m_3| + |m_2 - m_1|)} \quad (155)$$

Onde  $m_1, m_2, m_3$  e  $m_4$  representam as inclinações dos seguimentos  $\overline{12}$ ,  $\overline{23}$ ,  $\overline{34}$  e  $\overline{45}$ .

Sejam os pontos  $(x_1, y_1)$  e  $(x_2, y_2)$ , onde:

$$y = y_1 \quad e \quad \frac{dy}{dx} = t_1 \quad em \quad x = x_1 \quad (156)$$

$$y = y_2 \quad e \quad \frac{dy}{dx} = t_2 \quad em \quad x = x_2 \quad (157)$$

Para o cálculo da interpolação entre dois pontos, assume-se que a curva entre esses pontos pode ser representada por um polinômio de grau três, dado da seguinte forma:

$$y = p_0 + p_1(x - x_1) + p_2(x - x_1)^2 + p_3(x - x_1)^3 \quad (158)$$

Onde:

$$p_0 = y_1 \quad (159)$$

$$p_1 = t_1 \quad (160)$$

$$p_2 = \frac{\left[3\frac{(y_2 - y_1)}{(x_2 - x_1)} - 2t_1 - t_2\right]}{(x_2 - x_1)} \quad (161)$$

$$p_3 = \frac{\left[t_1 + t_2 - \frac{(y_2 - y_1)}{(x_2 - x_1)}\right]}{(x_2 - x_1)^2} \quad (162)$$

No fim da inclinação, dois pontos devem ser estimados e adicionados aos dados. Assim os pontos existentes  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  e os novos pontos  $(x_4, y_4)$  e  $(x_5, y_5)$  passam por uma curva representada pela seguinte equação.

$$y = g_0 + g_1(x - x_3) + g_2(x - x_3)^2 \quad (163)$$

Onde os parâmetros  $g$  são constantes. Assume-se que:

$$x_5 - x_3 = x_4 - x_2 = x_3 - x_1 \quad (164)$$

As ordenadas  $y_4$  e  $y_5$  podem ser determinadas a partir dos pontos  $x_4$  e  $x_5$ . Resultando na seguinte equação.

$$\frac{(y_5 - y_4)}{(x_5 - x_4)} - \frac{(y_4 - y_3)}{(x_4 - x_3)} = \frac{(y_4 - y_3)}{(x_4 - x_3)} - \frac{(y_3 - y_2)}{(x_3 - x_2)} = \frac{(y_3 - y_2)}{(x_3 - x_2)} - \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad (165)$$



### 3. ALGUNS TRABALHOS DESENVOLVIDOS

#### 3.1. Otimização Substituta

Gutmann (2001) apresentou um método para encontrar o mínimo global de uma função contínua em um subconjunto compacto de  $\mathbb{R}^d$ . Utilizou uma interpolação de base radial para definir uma função utilidade e concluiu que a convergência pode ser alcançada sem suposições adicionais sobre a função objetivo.

Wan, Pekny e Reklaitis (2005) apresentam uma estrutura de otimização baseada em simulações a partir da construção iterativa de um modelo substituto, baseado em resultados de simulações sistematicamente acumulados para capturar a relação causal entre as principais variáveis de decisão. As variáveis de decisão foram otimizadas usando o modelo substituto. Os resultados mostraram soluções eficientes com a utilização da estrutura de otimização.

Regis e Shoemaker (2007) apresentam um método de superfície de resposta estocástica proposto iterativamente. Este método utiliza um modelo de superfície de resposta e identifica um ponto promissor para a avaliação da função a partir de um conjunto de pontos gerados aleatoriamente. O método converge para o mínimo global e se torna uma abordagem promissora para a otimização global de funções complexas.

Couckuyt et al. (2009) discutem várias abordagens que utilizam otimização substituta e destacam uma abordagem sequencial de design em particular aplicada a problemas eletromagnéticos.

### 3.2. Modelos Substitutos

Ryu et al. (2002) descreveram em seu artigo as definições, as funções de estimativa e os algoritmos de dois tipos de kriging. O modelo de estimativa empregado foi o de interpolação a partir do DACE (Design and Analysis of Computer Experiments).

Martin e Simpson (2003) fizeram uma descrição básica sobre o Kriging destacando as principais semelhanças e diferenças entre os três principais tipos de Kriging (Simples, Ordinário e Universal), além disso aplicaram os três tipos em 6 estudos de caso diferentes e chegaram à conclusão que o uso de cada tipo de kriging pode ser melhor ou pior dependendo do caso aplicado.

Zhou e Turng (2007) estabeleceram um modelo substituto adaptativo com tempo de resposta curto e precisão adequada utilizando uma técnica de regressão estatística não linear e design de experimentos. Eles introduziram um modelo substituto especial baseado do processo gaussiano. Enquanto o modelo substituto é estabelecido, um algoritmo é empregado para avaliar o modelo e procurar as soluções globais ideais, trazendo resultados eficientes.

Liang e Song (2009) definiram dois estimadores restritos para os parâmetros de regressão em um modelo de regressão linear múltipla com erros de medição quando informações anteriores para os parâmetros são disponíveis.

Shahsavani e Grimvall (2011) propuseram um projeto adaptativo sequencial para derivar modelos substitutos e estimar índices de sensibilidade para diferentes subgrupos de entradas. Os pesquisadores defendem que o procedimento proposto se torna útil quando já pouco conhecimento prévio sobre a superfície de resposta e quando o objetivo é explorar a variabilidade global e características não lineares locais do modelo resultante.

Liu, Zhan e Tan (2012) desenvolveram um método de Kriging otimizado utilizando um algoritmo nomeado artifício de colônia de abelhas combinando amostragem de importância para problemas de confiabilidade estrutural. A metodologia se mostrou eficiente particularmente para problemas de alta não-linearidade, alta dimensionalidade e funções de desempenho implícitas.

Örkcü (2013) adaptou o algoritmo de regressão linear múltipla para um algoritmo heurístico híbrido. Ele concluiu que o método proposto pode ser uma alternativa eficiente aos métodos tradicionais de seleção de subconjunto para o problema de seleção de variável em modelos de regressão.

Regis (2015) desenvolveu um novo método de otimização baseado no kriging chamado TRIKE que implementa uma abordagem de região de confiança em que cada iteração é obtida maximizando uma função de melhoria esperada em alguma região de confiança.

Romero, Marim e Amon (2015) conduziram um experimento numérico para estudar a evolução de vários erros métricos durante o aprimoramento sequencial do modelo, estimar erros de predição e definir critério de parada adequado sem a necessidade de ter amostras adicionais além daquelas que já haviam sido utilizadas para a construção do modelo. Os resultados mostraram que é possível estimar o modelo com precisão se a utilização de amostras adicionais.

Para dados de entrada com grandes dimensões, o kriging se torna computacionalmente caro pois requer que a matriz seja invertida diversas vezes até que os parâmetros do modelo sejam estimados. Assim, Bouhlef et al (2016) melhoraram o kriging utilizando a construção de um núcleo de covariância que depende apenas de alguns parâmetros. Este núcleo foi construído com base em informações obtidas a partir do método de mínimos quadrados. Os resultados se mostraram satisfatórios para casos numéricos com até 100 dimensões.

Vicario, Craparotta e Pistone (2016) fizeram um estudo comparativo entre a utilização do kriging e as redes neurais, a fim de determinar qual modelo garantia uma melhor precisão na previsão do resultado de experimentos computacionais de dinâmica de fluidos, em quatro dimensões, para turbinas de baixa pressão onde são fornecidos valores de perda de energia. Verificou que o modelo de redes neurais é mais eficiente em relação à precisão na previsão dos resultados.

Yi (2016) utilizou o método do kriging para prever condições de luz do dia durante um ano inteiro e associar este método com ferramentas de simulação de energia. Com isto ele obteve resultados mais realísticos e conseguiu reduzir

o esforço computacional, comparado com os métodos que existiam anteriormente.

Kicsiny (2016) propôs um melhoramento no algoritmo de regressão linear múltipla na tentativa de minimizar o erro de modelagem. O modelo proposto utilizou uma regressão polinomial múltipla. Concluiu que o modelo é mais preciso, de fácil uso e com baixa demanda computacional.

Gaspar, Teixeira e Soares (2017) propuseram um modelo rigoroso do Kriging com refinamento ativo para resolver a avaliação de confiabilidade dos problemas, como nos casos de haver um único ponto de design, em funções com um número moderado de variáveis aleatórias de entrada. O modelo desenvolvido se mostrou eficiente.

Dirignei (2017) utilizou o método Kriging Multivariante aplicado a um sistema de suspensão de veículo, a partir de um algoritmo capaz de lidar com um grande número de parâmetros. O algoritmo proposto pode ser utilizado tanto para a estimação de probabilidade quanto para a validação.

Zhang et al. (2017) desenvolveu um modelo chamado método de convergência de linhas capaz de prever a função em um ponto onde a mesma não pode ser avaliada. O Kriging foi adotado para uma aproximação unidimensional, estimando não apenas o valor da função, mas também a incerteza da estimativa no ponto inacessível. O modelo mostrou ser preciso, robusto e confiável.

Wang, Wang e Zhao (2017) utilizaram o modelo de kriging no processo de otimização para atualizar o modelo FRF, um modelo baseado em função de resposta de frequência de aceleração. O kriging foi escolhido por ser um modelo rápido e de fácil aplicação.

Baareh (2019) utilizou dois modelos para testes, um modelo de rede neural “brack-propagation” e uma função de base radial. Concluiu-se que para as situações observadas, o modelo “back-propagation” obteve melhores resultados do que o modelo de função de base radial.

### **3.3. Técnicas de transformações de variáveis**

Wang (2008) utilizou a técnica de transformação de variáveis Boxcox para montar modelos estocásticos a partir de dados climáticos com o objetivo de contribuir para o planejamento de projetos de recursos hídricos. O método se mostrou eficaz na diminuição das incertezas e viabilidade computacional.

Gottardo e Raftery (2009) propuseram uma solução bayesiana para resolver o problema da transformação simultânea e seleção de variáveis para regressão. A solução permitiu calcular a média de todos os modelos considerados, incluindo transformações nas variáveis respostas e nos preditores. Para isso, utilizaram o método Boxcox.

Nguyen (2009) propôs um esquema eficaz utilizando uma abordagem de transformação de variáveis em combinação com o conceito de análise de componente independente com o objetivo de estimar com precisão a entropia de macromoléculas. O novo método se mostrou rápido, simples e produziu estimativas mais próximas da entropia exata do que as técnicas previamente disponíveis.

### **3.4. Validação Cruzada**

Xu, Liang e Du (2004) desenvolveram uma metodologia chamada de validação cruzada de Monte Carlo, a qual se mostrou consistente para a seleção do modelo. Com a utilização desta metodologia, é possível evitar um modelo muito grande e diminuir o risco de “overfitting” do modelo. Os resultados se mostraram satisfatórios.

Zhang e Wang (2010) aplicaram o método kriging juntamente com a validação cruzada. O método foi proposto pela eficiência da validação cruzada na previsão espacial de grandes dados para avaliar o quão bem uma aproximação funciona. Os testes foram feitos para determinação das concentrações de glicose de dados infravermelhos em amostras de mosto da produção de bioetanol e na modelagem do índice de retenção cromatográfica gasosa de compostos aromáticos policíclicos de descritores moleculares.

Filzmoser, Liebmann e Varmuza (2009) desenvolveram uma metodologia de validação cruzada dupla repetida para otimizar a complexidade dos modelos

de regressão e para estimar de forma realista os erros de previsão quando o modelo é aplicado a novos casos.

Bornn, Doucet e Gottardo (2010) mostraram que é possível utilizar métodos sequenciais de Monte Carlo para criar um algoritmo eficiente e automatizado. Eles demonstraram o algoritmo em um contexto de validação cruzada e utilizou para selecionar o parâmetros em regressões.

### **3.5. Técnicas de Amostragem**

Olsson, Sandberg e Dahlblom (2003) utilizaram diferentes versões de amostragem de hipercubo latino para melhorar o método de amostragem. Observaram que pode-se ter até 50% de economia computacional usando hipercubos latinos em vez do simples Monte Carlo.

Fang e Lin (2003) apresentaram a teoria e o método de design uniforme o qual se caracteriza por um tipo de projeto de experimento que pode ser usado para experimentos de computador e também para experimentos industriais quando o modelo subjacente é desconhecido.

Guerra et al. (2003) apresentaram o método de Voronoi para o cálculo de cargas atômicas, o qual se mostrou eficiente devido à sua simples partição geométrica do espaço. Os resultados encontrados pelo método apresentado foram satisfatórios quando comparados com experimentos químicos.

Chi, Mascagni e Warnock (2005) apresentaram um novo algoritmo para encontrar uma sequência de Halton ideal dentro de um espaço de embaralhamento linear. Esta sequência ideal foi testada numericamente e se mostrou melhor do que a sequência original. Eles também forneceram uma visão geral de vários algoritmos para a construção de várias sequências de Halton.

Wang et al. (2006) discutiram o projeto de experimentos D-ótimo para modelos de regressão de Poisson. Observaram que considerando o modelo de primeira ordem de uma variável, o projeto D-ótimo é independente dos parâmetros do modelo. Entretanto, em modelos mais complicados, o projeto D-ótimo depende dos parâmetros do modelo.

Du e Emelianenko (2006) estudaram novos algoritmos no método de amostragem Voronoi. Os estudos foram realizados a partir de análises teóricas e simulações computacionais e apresentaram resultados de convergência rigorosos. Concluíram que há uma redução significativa do esforço computacional quando os novos algoritmos foram comparados com os métodos tradicionais.

Moness, Linsley e Garzon (2007) utilizaram o método de amostragem da análise fatorial para determinar condições operacionais ideais em uma operação de moagem de aço. Foram testados diferentes projetos experimentais e concluíram que é importante ter bastante conhecimento para a escolha de um bom projeto experimental. Além de que, ter um projeto experimental de tamanho mínimo nem sempre resulta em descobertas produtivas.

Schlier (2008) apresentou vários tipos de embaralhamento para as sequências de baixa discrepância de Halton. Observaram que as sequências de deslocamento aleatório são menos eficientes em alguns casos e a sequência embaralhada não é mais eficiente do que uma sequência pseudoaleatória.

Zhou (2008) propôs um critério de design robusto D-ótimo para estudar projetos de superfícies de respostas. A variância e o viés foram considerados no projeto. O estudo comparou o projeto minimax D-ótimo com o projeto D-ótimo clássico. Verificou que o minimax D-ótimo se mostrou mais eficiente.

Sallaberry, Helton e Hora (2008) apresentaram um procedimento para estender o tamanho de uma amostra no hipercubo latino com variáveis correlacionadas de classificação. O procedimento destina-se ao uso em conjunto com a incerteza e análise de sensibilidade de modelos que exigem um grande esforço computacional em que é importante fazer uso eficiente de um número necessariamente limitado de avaliações do modelo.

Crombecq et al. (2008) fizeram uma comparação entre diferentes métodos de amostragens sequenciais para modelagem substituta global em um problema real. Propuseram ainda um critério de exploração utilizando um mosaico de Voronoi, cujos resultados indicaram uma melhoria considerável da precisão média do modelo.

Berrios et al. (2009) utilizou um projeto fatorial de 32 experimentos para encontrar as condições operacionais mais adequadas para a síntese de biodiesel a partir da banha de porco utilizando o hidróxido de potássio como catalisador. Os parâmetros estudados foram a concentração do catalisador e a velocidade de agitação.

Mannarswamy et al. (2009) utilizaram um projeto experimental D-ótimo para determinar modelos de isotermas de adsorção de Freundlich e Langmuir. A otimalidade do projeto foi verificada usando o teorema de equivalência geral.

Viana, Venter e Balabanov (2010) apresentaram um novo método para obter projetos de hipercubo latino ótimos sem a utilização de uma otimização formal. O novo método consiste em um algoritmo de propagação translacional. O algoritmo proposto representou uma estratégia atrativa do ponto de vista da economia computacional na obtenção de projetos de Latin Hipercubo de dimensões médias.

Yu, Goos e Vandebroek (2010) abordaram a técnica de amostragem tradicional seguindo uma abordagem pseudo Monte Carlo além de apresentarem outras abordagens como sequências de Halton, Faure, LHS. Fazendo as comparações entre as técnicas, observaram que todas as técnicas são eficientes para espalhar pontos em um espaço amostral.

Yahiaoui, Aissani-Benissad e Aït-Amar (2010) utilizaram a técnica de projeto fatorial de dois níveis para investigar parâmetros no processo de cimentação e puderam afirmar qual era o efeito mais influente. Os resultados experimentais foram aproximados por um modelo de segunda ordem.

Sinha e Xu (2011) discutiram a construção de projetos sequenciais D-ótimos para a análise de dados longitudinais usando modelos mistos lineares generalizados. Além da discussão, apresentaram um exemplo utilizando dados reais obtidos a partir de um estudo clínico.

Gurrieri (2011) investigou a perda de eficiência da sequência de Sobol quando testada em altas dimensões. Ele fornece também evidências numéricas de que é possível remover a maior parte desse viés e alcançar uma boa



convergência em dimensões altas se aplicar a randomização na sequência de Sobol.

Ke et al. (2012) utilizaram a técnica de amostragem baseada nas sequências de Hammersley e SVR para construir um modelo substituto a fim de reduzir custos computacionais. Observou-se que o SVR foi mais preciso e mostrou grande potencial de aplicação no projeto de tarefas complexas e computacionalmente caras.

Taherdoost (2016) apresenta etapas a se percorrer para conduzir uma amostragem, além de apresentar os diferentes tipos de técnicas e métodos de amostragem para ajudar os pesquisadores a coletar os dados amostrais de forma correta.

Chen et al. (2017) utilizaram a técnica de amostragem da sequência de Hammersley em um problema de otimização multiobjetivo com base no modelo de custo, eficiência elétrica e confiabilidade do fornecimento de energia híbrida, composta por sistemas de geração eólica, solar e células de combustível. Ao final, a pesquisa forneceu um método eficiente para os tomadores de decisão no projeto do sistema híbrido.

Bhattacharyya (2018) abordou uma comparação entre a técnica de design experimental de uma tentativa com número fixo de amostras e uma técnica de amostragem sequencial. As técnicas de amostragens foram aplicadas na construção de modelos substitutos mais precisos. O autor aborda duas técnicas de amostragem populares, como o latin hipercubo e a sequência de sobol e quatro técnicas de amostragem sequenciais. Os resultados apresentaram eficiência melhor quando comparados com a técnica clássica de amostragem de Monte Carlo.

### **3.6. Aprendizado de Máquina**

Devido às dificuldades de utilização com o software Auto Weka, Kothoff et al. (2016) desenvolveram uma nova versão denominada Auto Weka 2.0. Nesta versão, um sistema foi projetado para facilitar a utilização quando o usuário não compreende qual abordagem do aprendizado de máquina deve seguir de acordo com o seu conjunto de dados. No Auto Weka 2.0, um sistema foi projetado para

pesquisar automaticamente a melhor abordagem e suas respectivas configurações de hiperparâmetros para maximizar o seu desempenho a partir de um método de otimização bayesiana eficiente.

### **3.7. Outros estudos na área**

Gorissen et al. (2010) apresentaram um kit de ferramentas a qual reuniu algoritmos para ajuste de dados, seleção de modelo, seleção de amostra, otimização de hiperparâmetros e distribuição computacional. A ferramenta foi desenvolvida com o objetivo de capacitar um especialista a gerar de forma eficiente um modelo preciso para o seu problema em questão.

Sahidinis e Miller (2014) utilizaram uma metodologia que depende de um número reduzido de simulações para determinar os modelos. A metodologia tem início com a construção de um modelo substituto simples e é então melhorado sistematicamente através de solucionadores de otimização para adicionar de forma adaptativa novos pontos experimentais. A metodologia proposta recebeu o nome de ALAMO e se mostrou bastante eficaz tanto em relação ao modelo encontrado quanto na redução do esforço computacional.

Beaujean (2014) demonstrou como utilizar um estudo de Monte Carlo para decidir sobre o tamanho da amostra para a análise de regressão usando perspectivas de potência e precisão de parâmetro.

Boukouvala e Floudas (2015) apresentaram o algoritmo ARGONAUT para otimização de problemas gerais. Este algoritmo incorpora seleção de variáveis, limitação de limites e técnicas de amostragem com o objetivo de desenvolver modelos substitutos precisos. O algoritmo se mostrou eficiente em problemas de até 100 variáveis e 81 restrições, pois conseguiu encontrar a solução global utilizando menos amostras do que os métodos existentes. Entretanto, exige um custo computacional maior por resolver múltiplos problemas de otimização para estimativa de parâmetros.

Riley et al. (2018) propuseram um método para determinar o número mínimo de amostras. O valor mínimo de amostras deve respeitar quatro critérios específicos: Fator de redução global nas estimativas de efeito maior ou igual a 0,9; Coeficiente de correlação  $R^2$  menor ou igual a 0,05; Estimativa precisa do

padrão residual do desvio do modelo; Estimativa precisa do valor médio do resultado previsto. Encontrado o valor de  $n$  que respeite os quatro critérios anteriormente mencionados, este valor pode ser utilizado como o número mínimo de amostras para o problema. Este valor pode ser muito maior a depender da complexidade.

Jenkins e Quintana-Ascencio (2020) estudaram a influência de diferentes tamanhos de amostras para estimativa de parâmetros de regressão e metaregressão a fim de encontrar um número mínimo ideal que diminuísse a variância. Eles concluíram que um número mínimo aceitável de amostras para determinar parâmetros de regressão deve ser igual ou superior a 25, a depender da complexidade do problema.

#### **4. METODOLOGIA**

Este trabalho propõe um algoritmo que seja capaz de construir modelos substitutos ou metamodelos de forma automática. Neste algoritmo, o usuário não precisará informar o número de amostras necessário ou o metamodelo que será utilizado. Como resposta, o algoritmo fornecerá ao usuário o número de amostras necessárias para a construção do modelo, as transformações das variáveis respostas e o melhor modelo que representa os dados.

As variáveis respostas são as variáveis de interesse observadas durante as simulações. Neste algoritmo, as variáveis respostas podem ser classificadas em três grupos: ativo, convergido e não convergido. No início, todas as variáveis são consideradas ativas, pois ainda não começaram a ser processadas. Em seguida, o algoritmo seleciona uma variável ativa, aqui chamada “kresp”, (o critério será explicado posteriormente) e utiliza uma função de otimização para encontrar o seu modelo. Quando o algoritmo encontra a solução para uma variável resposta, esta variável vai para o conjunto “convergido”. Caso o algoritmo não encontre o modelo para a variável “kresp”, esta vai para o grupo “não convergido”. No background, todas as variáveis do conjunto “ativo” e do conjunto “não convergido” também estão sendo processadas sem a utilização da função de otimização.

Este algoritmo utiliza um esquema de controle do tipo proporcional para apresentar uma solução mais eficiente do que a utilização de um número fixo de amostras. O número de amostras ( $u$ ) em cada iteração funciona como a variável manipulada (MV) e deve obedecer os limites de 1 até um limite máximo definido pelo usuário. A variável controlada (CV) é um parâmetro, aqui chamado, de “Erro”. Este parâmetro permite identificar que o erro atingiu um valor mínimo aceitável e praticamente em estado estacionário, o qual pode ser calculado pela equação a seguir.

$$Erro_r = \frac{|\Delta Erro_i|}{u_{i-1}} = \frac{|Erro_i - Erro_{i-1}|}{u_{i-1}} \quad (166)$$

Onde, Erro é apresentado por Wilson e Sahidinis (2017) e mostrado na seguinte equação.

$$Erro = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (167)$$

E,  $i$  corresponde ao índice da iteração;  $u$  corresponde ao número de amostras que foram adicionadas naquela iteração correspondente,  $\bar{y}$  corresponde à média dos dados de respostas de treinamento.

O “setpoint” do controlador é definido pelo valor zero, o qual indica que não haverá mais nenhum progresso durante a simulação. A escolha desta CV está no fato de que esta variável indica que não há mais variação significativa no erro durante o progresso da iteração.

Este critério poderia ter sido feito utilizando apenas a diferença do erro, conforme mostrado na equação a seguir:

$$\Delta Erro = |Erro_i - Erro_{i-1}| \quad (168)$$

Nota-se que a diferença entre a Equação (168) e a Equação (166) é a ausência do denominador, representado pelo número de amostras naquela iteração. Na ausência do denominador, o valor do erro poderia ser razoavelmente grande, o qual iria interferir no cálculo do ganho do controlador.

Para detectar que o algoritmo atingiu o estado estacionário, foi aplicada a técnica de detecção do estado estacionário proposta por Rhinehart (2013) e apresentada na seção 2.8 deste trabalho.

Um outro parâmetro que será utilizado para verificar a aderência dos dados ao modelo é a raiz do erro quadrático médio, representado pela variável  $Q^2$  e calculado pela expressão  $1 - Error$ .

Para chegar ao valor do erro, o algoritmo passa por uma sequência de etapas:

1. Aplicação das técnicas de transformação ou expansão das variáveis;
2. Escolha do melhor modelo de regressão (linear/não linear);
3. Validação cruzada;
4. Cálculo do Erro e  $Q^2$ .

A determinação do melhor metamodelo pode ser encontrada por três tipos de otimização: Força bruta (todas as possíveis combinações), otimização substituta e otimização bayesiana.

Os vinte e cinco metamodelos disponíveis no algoritmo são do tipo “kriging” e são eles: Squared exponential; ARD Squared Exponential; Rational Quadratic; ARD Rational Quadratic; Matern12; Matern32; Mater52; ARD Matern12; ARD Matern32; ARD Matern52; Neural Network; Piecewise Polynomial; ARD Piecewise Polynomial; Gabor; ARD Gabor; Periodic Squared Exponential; ARD Periodic Squared Exponential; Periodic Rational Quadratic; ARD Periodic Rational Quadratic; Periodic Matern12; Periodic Matern32; Periodic Matern52; ARD Periodic Matern12; ARD Periodic Matern32; Periodic Matern52.

Os principais metamodelos já foram descritos na Seção 2.6.

O algoritmo é, então, apresentado a seguir. O algoritmo é considerado inclusivo porque todas as variáveis que não atenderam aos critérios de saída para convergência são processadas de alguma forma, mesmo aquelas que foram consideradas “não convergidas” de acordo com os mesmos critérios de saída.

---

**Algoritmo** “Algoritmo heurístico de retroalimentação inclusiva para regressão de dados de processo”

**Início**

---

---

```

//Comunicação entre o simulador gerador de dados e o
algoritmo
//Aspen Plus
Leia //Arquivo Aspen Plus
Leia //Variáveis de entrada e saída de acordo com a
convenção de linguagem Aspen Plus
Leia xmin, xmax //Limites inferiores e superiores para
as variáveis de entrada
Leia rangeY //Imagem de cada variável resposta
//Simulink
Leia //Arquivo Simulink
//O usuário deve procurar a função simulFuncSimulink
Leia //Variáveis de entrada e saída
Leia sim_time //Tempo de simulação em segundos
Leia optionsSimulink
Leia xmin, xmax // Limites inferiores e superiores
para as variáveis de entrada
Leia rangeY //Imagem de cada variável resposta
//Função ou script do Matlab
//Abrir a função SimulFuncMatlab
Leia //Variáveis de entrada e saída
Leia xmin, xmax //Limites inferiores e superiores para
as variáveis de entrada
Leia rangeY //Imagem de cada variável resposta
//Configuração de estrutura, método e objeto do
projeto sequencial básico
Crie struct
Escolha methods
    Caso intersite-projected
    Caso intersite-projected-threshold
    Caso optimizer-projected
    Caso optimizer-interside
Fimescolha
Crie seq //Objeto de design sequencial
Leia iniModelLI, iniModelNL, kresp, ModelNL,
maxSamplesPerResp, perctSamplesLinear, ceil, islola,
kfold, maxTime, lbQ2, maxDu, minKc, nss, optMethod, Mmax
//Parâmetros que podem ser modificados pelo usuário. Para
cada variável dessa é assumido um valor padrão
//Cálculo de parâmetros internos
maxSamplesLinear ← ceil(perctSamplesLinear *
maxSamplesPerResp)
Se perctSamplesLinear ← 0 Então
    Use kresp ← ModelNL
    Use iniModelNL //para as outras variáveis ativas
Senão
    Use iniModelLI //para as outras variáveis ativas
Se samplesPerResp < maxSamplesLinear Então
    Use kresp ← iniModelLI
Senão
    kresp ← ModelNL

```

---

---

```

Fimse
Fimse
//Escolher se as variáveis de entrada serão
transformadas ou expandidas
Escolha par.flagTransformation
    Caso true
    Caso false
Fimescolha
Escolha par.flagExpansion
    Caso linear //sem expansão
    Caso interaction
    Caso quadratic
    Caso purequadratic
    Caso //outra expansão implementada pelo usuário
Fimescolha
Inicialize VTrans //e demais matrizes
m ← max{[(kfold/(kfold-1)) * "inputs"],2*kfold}
//geração de pontos iniciais
Função SimulFuncXXX(points) //Simulações com os
pontos gerados
    Retorne y //Variáveis respostas
Fimfunção
Para todas as respostas do grupo "ativo" faça
    //Cálculo do erro
Fimpara
u ← 1
//gere u pontos utilizando o Sequential Design Method
// Obtenha as respostas para os pontos gerados
Para todas as respostas do grupo ativo faça
    //Cálculo do erro
Fimpara
Erro_rn ← abs(erro_n - erro)
Erro_n ← Erro
Enquanto conjunto "ativo" ≠ 0 & tempo < maxTime faça
    u ← 2
    Para todas as respostas do conjunto "ativo" e
"não convergido" faça
        //Cálculo do Erro e Q2
Fimpara
Selecione kresp //menor Q2
Se islola ← true então
    //Projeto Lola-Voronoi é configurado para
selecionar novos pontos com base nas variáveis respostas
de "kresp"
Fimse
Enquanto critério kresp ainda não for atingido &
samplesPerResp < maxSamplesResp & tempo < maxTime faça
    Se samplesPerResp ≥ maxSamplesLinear Então
        Use kresp ← ModelNL
Fimse

```

---

---

```

para kresp //Cálculo dos parâmetros do controlador
Kp ← abs(Erro_rn(kresp) - Error(kresp))/u
Kc ← 1/Kp

Erro_rn ← Error
u ← 0 + Kc(Error(kresp) - 0) //Lei de
controle para determinar a quantidade de novos pontos
(inteiro positivo)
//Gere u pontos (método de amostragem
escolhido no início do algoritmo ou Lola-Voronoi, se
estiver ativo.
Erro_n ← Erro
Para todas as respostas do grupo ativo
(exceto kresp) e do grupo não convergido faça
//Cálculo do erro e Q2
Fimpara
Para resposta ativa "kresp" faça
Função varTransModelOptimization
Retorne Modelo de regressão,
funções de transformações das variáveis respostas, método
de regressão e função de covariância, técnica de
escalonamento, Erro, Q2.
Imprima Resultados e gráficos
Fimfunção
Fimpara
Se número de iterações >= nss Então
Para todas as respostas do conjunto
"ativo" faça
//Verifique se as respostas
atingiram o estado estacionário (critério de saída 1)
//Verifique se o valor do Q2
resultante da interpolação linear do estado estacionário
está acima de "lbQ2" (critério de saída 2)
//Verifique se o valor de Q2 está
acima de "lbQ2" por "nss" iterações consecutivas (critério
de saída 3)
Fimpara
Fimse
Atualize samplesPerResp e o tempo
processado
Fimenquanto
//Caso "kresp" ultrapasse o número máximo de
amostras sem atingir os critérios de saída, ela irá para o
grupo não convergido.
Erro_n ← Erro
Erro_rn ← Erro_r
Fimenquanto
Imprima Manipulações das variáveis regressoras e
respostas, seleção de preditores adequados, metamodelo
final, dados da validação cruzada.

```

---



---

## **Fimalgoritmo**

---

A seguir, será apresentada uma descrição do passo a passo de como este algoritmo funciona para que seja possível fornecer as melhores soluções para o usuário.

### **4.1. Comunicação do simulador gerador de dados com o algoritmo**

O algoritmo funciona em comunicação direta com o simulador gerador de dados durante toda a etapa de construção dos metamodelos. A comunicação pode ser feita com os simuladores Aspen Plus e Simulink, além de poder inserir os dados por meio de uma função de Matlab ou script.

Para o Aspen Plus, o usuário deve inicialmente indicar o arquivo gerador de dados (simulação), as variáveis de entrada e saída que auxiliarão na construção dos modelos, os limites inferiores e superiores para as variáveis de entrada e a imagem para cada uma das variáveis de saída.

Vale ressaltar que a definição das variáveis deve estar de acordo com a convenção de linguagem Aspen Plus. Considerando o todo, há três opções para o “rangeY” que as variáveis respostas podem assumir: Apenas valores positivos; apenas valores negativos; valores positivos e negativos. Na maioria das variáveis de simulações de processos, as imagens das variáveis de saída são subconjuntos de  $\mathbb{R}^+$ , devido à natureza das variáveis de processo químico.

Ramirez e Antonio (2018) apresentaram o procedimento completo de comunicação entre os softwares Matlab e Aspen Plus. O primeiro passo é ter a simulação de interesse em Aspen Plus, na extensão .bkg. Deve-se então, identificar os objetos e as restrições do problema de otimização e suas variáveis envolvidas. Essas variáveis devem ser declaradas no código de otimização e serão usados para atribuir valores do MATLAB no arquivo em Aspen Plus. A partir daí, deve-se dar o comando para que o MATLAB abra a simulação em Aspen Plus através do “actxserver”.

Com a simulação aberta, o Matlab envia as informações necessárias para realizar a simulação e o comando para que a simulação seja realizada. Quando

finalizada, o Matlab recolhe os dados para continuar com a estratégia da otimização.

Para o Simulink, o usuário deve inicialmente indicar o arquivo responsável pela geração de dados e, para indicar quais são as variáveis de entrada e saída construídas no modelo, o usuário irá procurar pela função “simulFuncSimulink” e definir quais são essas variáveis. Esta função será apresentada em detalhes no decorrer da metodologia.

Em seguida, o usuário deve retornar ao arquivo principal e definir o tempo de simulação. Nesta etapa, o usuário precisa ter domínio da engenharia do processo para garantir que o tempo definido seja longo o suficiente, capaz de permitir que a simulação alcance o estado estacionário.

Por fim, deve definir as opções de simulação e a imagem de cada uma das variáveis de saída. Esta última etapa é feita de forma equivalente quando utilizado o simulador Aspen Plus.

Para inserir os dados por meio de uma função ou script no Matlab, o usuário deve, primeiramente, abrir o arquivo “simulFuncMatlab” e inserir as variáveis de entrada e saída neste arquivo. Em seguida, deve definir a imagem de cada uma das variáveis de saída, de forma equivalente quando utilizados os simuladores Aspen Plus e Simulink.

#### **4.2. Definição da estrutura do problema e seleção do método de amostragem sequencial**

Nesta etapa, o algoritmo criará uma variável “struct”, a qual conterá a quantidade de variáveis de entrada e as informações dos limites inferiores e superiores das mesmas.

O método de amostragem sequencial permite a utilização de amostras com tamanho variável, o oposto da amostragem convencional, a qual utiliza um número fixo de unidades amostrais.

Existem vários métodos de amostragem sequencial na literatura. No algoritmo apresentado, o usuário pode escolher entre quatro diferentes métodos

que são: “Intersite-projected”; “Intersite-projected-threshold”; “Optimizer-projected” e “Optimizer-intersite. Estes modelos já foram descritos na Seção 2.2.2.

Baseado em pesquisas na literatura, dentre os possíveis métodos de amostragem sequencial que o usuário pode escolher no algoritmo, o mais indicado é o “Intersite-projected-threshold”. Neste método de amostragem sequencial, a distância projetada mínima é especificada e o número de amostras tende a diminuir, isto porque, se houver muitos pontos próximos uns dos outros, alguns poderão ser descartados (BHATTACHARYYA, 2018).

O “intersite-projected-threshold” se mostra eficiente em relação à quantidade de respostas previstas mesmo para problemas de alta dimensão. Para Bhattacharyya (2018), a utilização deste tipo de amostragem, quando comparada com os outros tipos, previu melhores resultados na construção de modelos do tipo kriging, considerando o parâmetro de minimização do erro local.

### **4.3. Definição de alguns parâmetros**

Para encontrar os modelos, o algoritmo precisa da definição de valores de alguns parâmetros. O algoritmo traz valores padrão. Entretanto, o usuário pode fazer modificações a depender do seu conhecimento do problema com o objetivo de melhorar a convergência da construção dos modelos.

Os valores padrão para a escolha dos metamodelos serão apresentados nos resultados. Os parâmetros são apresentados a seguir:

- “iniModelLI”: Escolha do método de regressão linear para as respostas ativas.
- “iniModelNL”: Escolha do método de regressão não linear para as respostas ativas (exceto a resposta “kresp”) e escolha do método usado pelas respostas não convergentes nos cálculos de “background”.
- “ModelNL”: Conjunto de funções de covariância (métodos de regressão não linear) para regressão de processos gaussianos usando a caixa de ferramentas GPR do Matlab ou pacote GPML.

- “maxSamplesPerResp”: Número máximo de amostras usadas para processar cada resposta.
- “perctSamplesLinear”: Porcentagem do número máximo de amostras por resposta (“maxSamplesPerResp”) sendo processada por regressão linear. Em última análise, esse parâmetro é responsável por determinar a estrutura dos modelos finais. Se o usuário definir o valor deste parâmetro como sendo igual a zero, significa que o método escolhido será o método não linear para processar todas as respostas. Se o usuário definir o valor desta variável como sendo igual a 1, o método de regressão linear será o método escolhido para todas as respostas.
- “islola”: Define se o algoritmo utilizará usando o método Lola-Voronoi na geração de pontos para iterar a resposta ativa “kresp”.]
- “kfold”: Número de grupos para a validação cruzada. Este parâmetro enfatiza a previsibilidade aumentando o tamanho do conjunto de teste. Significa que  $1/kfold$  de dados será usado para validação, que deve ser suficientemente grande para reforçar a capacidade preditiva.
- “maxTime”: Tempo máximo de processamento. Deve ser definido em segundos.
- “lbQ<sup>2</sup>”: Limite inferior do Q<sup>2</sup>.
- “maxDu”: Número máximo de pontos adicionados durante a iteração.
- “minKc”: Ganho mínimo do controlador para garantir que o incremento “du” seja grande o suficiente.
- “nss”: Número de iterações consecutivas necessárias para definir o critério de parada do algoritmo.
- “optMehod”: Método de otimização utilizado na resposta “kresp”. Se o usuário definir o número 1, utilizará todas as combinações possíveis. Se definir o número 2 utilizará a otimização de substitutos. Pode ainda escolher a otimização Bayesiana 3 ou 4 dependendo do método de regressão (linear ou não linear) e do número de variáveis de decisão.

- “Mmax”: Número máximo de avaliações de função para cada reinicialização de otimização.

Estes parâmetros serão utilizados para calcular outros parâmetros internos usados pelo algoritmo.

#### 4.4. Cálculo de alguns parâmetros internos

Nesta etapa, o algoritmo irá calcular alguns parâmetros internos que serão utilizados no processo de construção dos modelos. Aqui será definido o número máximo de amostras utilizadas na regressão linear para cada resposta em processamento. Este valor será determinado pela seguinte expressão:

$$\text{maxSamplesLinear} = \text{ceil}(\text{perctSamplesLinear} \times \text{maxSamplesResp}) \quad (169)$$

Onde: “perctSamplesLinear” representa a porcentagem do número máximo de amostras por resposta sendo processada por regressão linear. Esse parâmetro irá designar se o método de regressão utilizado será o linear ou não linear e “maxSamplesResp” representa o número máximo de amostras usadas para processar cada resposta.

Se o usuário escolher a porcentagem do número de amostras processadas por regressão linear como sendo igual a zero ( $\text{perctSamplesLinear} = 0$ ), então o método de regressão para a resposta ativa “kresp” será o modelo não linear (ModelNL) e o método de regressão para todas as outras respostas ativas será o modelo não linear (iniModelNL).

Caso o parâmetro ( $\text{perctSamplesLinear} \neq 0$ ), o método de regressão usado para todas as variáveis ativas, exceto “kresp” será o modelo linear (iniModelLI). Se o número de amostras para cada “kresp” ativa for menor do que o número máximo de amostras utilizadas na regressão linear ( $\text{samplesperResp} < \text{maxSamplesLinear}$ ), então o método de regressão utilizado para a resposta ativa “kresp” será o método linear (iniModelLI). Caso ( $\text{samplesperResp} > \text{maxSamplesLinear}$ ), então o método de regressão utilizado para a resposta ativa “kresp” será o método não linear (ModelNL).

As variáveis respostas pertencentes ao grupo “não convergido” serão sempre processadas utilizando o método de regressão linear dos mínimos quadrados e um método de regressão não linear.

#### 4.5. Manipulação das variáveis de entrada e saída

Nesta parte do algoritmo, o usuário precisará escolher qual tipo de manipulação será realizada nas variáveis de entrada. Essa manipulação pode ser transformação ou expansão das variáveis.

O primeiro comando é o “par.flagTransformation” que determina se as variáveis de entrada serão transformadas ou não. Caso o usuário escolha como “true”, o algoritmo usará funções elementares para transformar as entradas gerando um conjunto de preditores de maior dimensão. Se escolher “false”, os preditores serão os mesmos que as variáveis de entrada.

Conforme descrito anteriormente na Seção 2.3, a técnica de transformação das variáveis permite reduzir a variância dos resíduos e assim, obter um metamodelo mais acurado.

Neste algoritmo são sugeridas dezoito transformações de variáveis, estas transformações não afetam os possíveis preditores que poderão assumir valores negativos. O usuário tem a liberdade de modificar estas transformações, se assim desejar.

Seja  $x$  o vetor que contém as variáveis regressoras  $[x_1, x_2, x_3, \dots, x_n]$ , as dezoito transformações (T) das variáveis sugeridas neste algoritmo são mostradas nas equações a seguir e todos os regressores pertencem ao domínio  $\mathbb{R}$ .

$$T_1 = x \quad (170)$$

$$T_2 = x^2 \quad (171)$$

$$T_3 = x^3 \quad (172)$$

$$T_4 = \sqrt[3]{x} \quad (173)$$

$$T_5 = \sqrt[5]{x} \quad (174)$$

$$T_6 = \sqrt[3]{x^2} \quad (175)$$

$$T_7 = \sqrt[5]{x^2} \quad (176)$$

$$T_8 = \sqrt[5]{x^3} \quad (177)$$

$$T_9 = \frac{1}{x} \quad (178)$$

$$T_{10} = \frac{1}{x^2} \quad (179)$$

$$T_{11} = \sqrt{|x|} \quad (180)$$

$$T_{12} = \ln |x| \quad (181)$$

$$T_{13} = \text{sen } x \quad (182)$$

$$T_{14} = \text{cos } x \quad (183)$$

$$T_{15} = \text{tan } x \quad (184)$$

$$T_{16} = \text{sen } \frac{1}{x} \quad (185)$$

$$T_{17} = \text{senh } \frac{x}{\alpha_1} \quad (186)$$

$$T_{18} = e^{\frac{x}{\alpha_2}} \quad (187)$$

Os parâmetros  $\alpha_1$  e  $\alpha_2$  devem ser definidos pelo usuário para garantir que as respectivas funções de transformação não atinjam valores muito altos.

O segundo comando é o “par.flagExpansion” e ele se refere à expansão das variáveis de entrada feita pelo recurso “x2fx” do Matlab. O usuário pode escolher entre “linear”, “interaction”, “quadratic”, “purequadratic”, ou ainda uma expansão implementada por ele. O recurso “x2fx” converte uma matriz de preditores em uma matriz de projeto para análise de regressão.

De acordo com Mathworks (2019), se  $X$  tem  $n$  colunas, a ordem das colunas da matriz de projeto para um modelo quadrático completo é o termo constante, os termos lineares (as colunas de  $X$  na ordem  $1, 2, \dots, n$ ), os termos de interação (pares dos produtos das colunas de  $X$ , em ordem,  $(1,2), (1,3), \dots, (1,n), (2,3), \dots, (n-1,n)$ ), os termos quadráticos (em ordem

1, 2, ..., n). Outros modelos usam um subconjunto desses termos, na mesma ordem.

Para as variáveis respostas, o algoritmo apresenta 27 possíveis transformações elementares, todas invertíveis, estas transformações são apresentadas a seguir.

$$T_1 = y \quad (188)$$

$$T_2 = y^3 \quad (189)$$

$$T_3 = y^5 \quad (190)$$

$$T_4 = y^7 \quad (191)$$

$$T_5 = \sqrt[3]{y} \quad (192)$$

$$T_6 = \sqrt[5]{y} \quad (193)$$

$$T_7 = \sqrt[7]{y} \quad (194)$$

$$T_8 = \sqrt[5]{y^3} \quad (195)$$

$$T_9 = \sqrt[7]{y^3} \quad (196)$$

$$T_{10} = \sqrt[7]{y^5} \quad (197)$$

$$T_{11} = \frac{1}{y} \quad (198)$$

$$T_{12} = \frac{1}{y^3} \quad (199)$$

$$T_{13} = \frac{1}{y^5} \quad (200)$$

$$T_{14} = \frac{1}{y^7} \quad (201)$$

$$T_{15} = \sqrt[3]{\frac{1}{y}} \quad (202)$$



$$T_{16} = \sqrt[5]{\frac{1}{y}} \quad (203)$$

$$T_{17} = \sqrt[7]{\frac{1}{y}} \quad (204)$$

$$T_{18} = \sqrt[5]{\frac{1}{y^3}} \quad (205)$$

$$T_{19} = \sqrt[7]{\frac{1}{y^3}} \quad (206)$$

$$T_{20} = \sqrt[7]{\frac{1}{y^5}} \quad (207)$$

$$T_{21} = \operatorname{senh} \frac{z}{\alpha_1} \quad (208)$$

$$T_{22} = \frac{1}{\operatorname{senh} \frac{z}{\alpha_1}} \quad (209)$$

$$T_{23} = \sqrt{|z|} \quad (210)$$

$$T_{24} = \sqrt{|z|^3} \quad (211)$$

$$T_{25} = \sqrt{|z|^5} \quad (212)$$

$$T_{26} = \sqrt{|z|^7} \quad (213)$$

$$T_{27} = \ln |z| \quad (214)$$

O parâmetro  $\alpha_1$  deve ser definido pelo usuário, assim como nas transformações das variáveis de entrada, para garantir que elas não atinjam valores muito altos. O intervalo de cada função de transformação deve ser a linha real  $\mathbb{R}$  para garantir que a transformação inversa exista. Por este motivo, deve-se ter cuidado na escolha dessas funções de transformações e não inserir funções que pertençam apenas ao intervalo  $\mathbb{R}^+$  ou  $\mathbb{R}^-$ .

#### 4.6. Inicialização de matrizes

Neste momento, o algoritmo fará a inicialização das matrizes do projeto. Particularmente a matriz “vTrans” das variáveis de decisão para otimização deve ser inicializada. Os elementos dessa matriz são, em ordem, o tipo de transformação das variáveis de resposta, o método de regressão/função de covariância e o método de escalonamento, para cada resposta. Neste ponto, o grupo “ativo” contém todas as variáveis e o grupo “não convergido” encontra-se vazio.

#### 4.7. Geração de pontos iniciais (m)

Para que o algoritmo possa realizar os cálculos e as operações iterativas, ele precisa gerar uma quantidade de pontos iniciais. O algoritmo irá espalhar os pontos iniciais conforme o método escolhido na etapa 4.2.

O tamanho da amostra deve garantir um mínimo de dois elementos em cada grupo (fold) para validação cruzada a fim de evitar valores  $-Inf$  para  $Q^2$ . Então o número de pontos iniciais será dada pela maximização da função a seguir.

$$m = \max\left\{\left\lceil \frac{kfold}{kfold-1} \times \text{"ninputs"} \right\rceil, 2 \times kfold\right\} \quad (215)$$

Onde kfold, é o número de grupos da validação cruzada (definido pelo usuário) e “ninputs” é o número de variáveis de entrada ou variáveis regressoras. Vale ressaltar que sempre que novos pontos forem gerados, a matriz de projeto das entradas deve ser verificada quanto a presença ou não de “zeros”, por meio da função “checkpoint”. Se houver algum “zero”, ele é substituído por um valor mais próximo.

De posse dos pontos iniciais, deve-se realizar as simulações desses pontos e coletar as informações das variáveis que serão observadas, ou seja, as variáveis respostas.

#### **4.8. Cálculo do valor inicial do “Erro” para todas as respostas (Erro<sub>rn</sub>)**

Para que seja possível calcular o valor do ganho do controlador, o algoritmo precisa de alguns valores iniciais. Nesta etapa, o algoritmo irá calcular um valor inicial de Erro<sub>rn</sub> para todas as respostas.

Conforme mostrado na Equação (166), para o cálculo do Erro são necessárias duas iterações. Assim, para se ter um valor inicial do Erro, é acrescentado mais uma amostra ao conjunto ( $u=1$ ).

Ao final desta etapa do algoritmo, o valor do erro inicial pode ser calculado respeitando a equação já mostrada anteriormente, onde nesta situação, o valor de  $u$  é igual a 1, pois foi acrescentada uma amostra ao conjunto de amostras existente.

Este desvio “Erro<sub>rn</sub>” será usado como variável controlada no esquema tipo feedback-control para definir o número de novos pontos a serem adicionados.

#### **4.9. Cálculo valor do Erro<sub>r</sub>**

Esta etapa é importante para encontrar os valores essenciais dos parâmetros “Kp” e “Kc” do controlador. Este controlador será útil adiante para a determinação do número de amostras que devem ser acrescentadas. Nesta etapa mais duas amostras são adicionadas ao conjunto de amostras existentes e o valor de Erro<sub>r</sub> é calculado a partir da Equação (166).

Neste ponto, o algoritmo também calcula o valor do  $Q^2$ . O cálculo desses dois parâmetros é realizado para todas as respostas do grupo “ativo” e “não convergido”.

#### **4.10. Escolha da resposta que será processada (kresp)**

Neste ponto, o algoritmo fará um “loop” para encontrar os melhores modelos para todas as respostas do processo. Para ajudar a encontrarmos o menor número de amostras possíveis, o algoritmo começa a escolha do melhor metamodelo para a resposta que é mais difícil, isto porque, algumas respostas

consideradas mais fáceis podem ser resolvidas enquanto o algoritmo tenta resolver a resposta escolhida para processamento.

A escolha da resposta mais difícil se baseia no valor do parâmetro  $Q^2$  para cada resposta, calculado anteriormente. O algoritmo considera a resposta mais difícil como sendo aquela que apresenta o menor valor para  $Q^2$ , já que este parâmetro indica o progresso do algoritmo a cada iteração para aquela resposta que está sendo processada.

Se o método Lola-Voronoi estiver ativo para ser utilizado, serão selecionados novos pontos baseados nas variáveis respostas. Observação: Neste ponto, o algoritmo aumenta o número máximo de amostras utilizadas na primeira “kresp” escolhida, isso é feito para compensar o número máximo de iterações consecutivas necessárias para definir o critério de parada (nss).

#### 4.11. Cálculo dos parâmetros do controlador

Para definir a quantidade de número de amostras que serão adicionadas, é necessário calcular os parâmetros do controlador  $K_p$  e  $K_c$ . O ganho do processo  $K_p$  corresponde ao Erro para a resposta “kresp” e será calculado pela seguinte equação:

$$K_p = \frac{|Erro_m(kresp) - Erro_r(kresp)|}{u} \quad (216)$$

Onde  $u$  corresponde ao número de amostras que foram adicionadas na última iteração, no caso desta etapa, duas amostras.

O ganho do controlador  $K_c$  é calculado pela equação a seguir:

$$K_c = \frac{1}{K_p} \quad (217)$$

#### 4.12. Cálculo do número de amostras que serão acrescentadas

A quantidade de amostras que serão adicionadas,  $u$ , à quantidade de amostras existentes será calculada pela lei de controle, apresentada a seguir:

$$u = 0 + K_c(Erro_r(kresp) - 0) \quad (218)$$

O objetivo deste esquema tipo feedback-control é conduzir o “Erro” da resposta “kresp” para o estado estacionário, não necessariamente o ideal igual a zero. É o equivalente a atingir o  $Error(kresp)$  igual a zero. Este valor é o ideal porque quando não há Erro, nenhum ponto é adicionado. O número de amostras “u” deve ser arredondado de forma que seja sempre um inteiro positivo e respeitar o intervalo  $u \in [1, maxu]$ .

A distribuição destas novas amostras será feita utilizando o método de amostragem sequencial escolhido no início do algoritmo ou o método Lola-Voronoi, se estiver ativo. Após a distribuição, são feitas simulações com os novos pontos para obtenção das variáveis respostas.

#### **4.13. Construção dos metamodelos para as respostas**

As variáveis dos grupos “ativo”, exceto “kresp”, e do grupo “não convergido” são processadas em background. Durante esses cálculos, as variáveis de decisão para otimização (tipo de transformação da resposta, método de regressão/função de covariância e método de escalonamento) são fixadas em seus valores nominais para as respostas ativas e, para as respostas não convergidas, são fixados em seus últimos valores quando estas respostas eram ativas.

A seleção de regressores é sempre realizada pela rotina de seleção de variáveis sequenciais baseada em OLS. A seleção do modelo de regressão nos cálculos de “background” é determinada pelo algoritmo e pode ser linear ou não linear. Se for linear, somente a regressão linear ordinária é executada por meio da rotina de seleção de variáveis baseada em OLS e compara o respectivo  $Q^2$  com o do modelo de regressão não linear, o maior valor definirá o modelo de regressão final.

A resposta ativa “kresp” será processada usando o método de otimização inteira não linear com a função “varTransModelOptimization”. Após o método ter sido encontrado, a função retornará ao usuário: o número de amostras, as funções de transformações das variáveis respostas, método de regressão/função de covariância, método de escalonamento, valor do Erro e  $Q^2$ .

A mesma técnica de escalonamento é aplicada para as variáveis de entrada e saída. No algoritmo, existem quatro métodos disponíveis, apresentados a seguir.

- “No scaling”: Usa as variáveis regressoras brutas e os dados de resposta;
- “standardization”:

$$\frac{z-\bar{z}}{s_z} \quad (219)$$

Onde  $\bar{z}$  e  $s_z$  representam a média o desvio padrão do argumento vetorial.

- “normalization”:

$$\frac{z-z_{min}}{z_{max}-z_{min}} \quad (220)$$

Onde  $z_{max}$  e  $z_{min}$  representam os valores máximos e mínimos do argumento vetorial.

- “Euclidean normalization”:

$$\frac{z-\bar{z}}{\|z-\bar{z}\|_2} \quad (221)$$

Onde o denominador da equação representa a distância euclidiana de norma 2.

O termo referente ao método de regressão ou função de covariância depende da condição a qual o algoritmo está sendo processado. Para as respostas do grupo “não convergido”, este termo será sempre um modelo de regressão não linear com uma função de covariância. Para as respostas do conjunto “ativo” processando sob regressão linear, este elemento é sempre o método de regressão dos mínimos quadrados. Se estiver sendo processado sob regressão não linear, será sempre baseado na regressão por processo Gaussiano (GPR), com uma das 25 funções de covariância apresentadas anteriormente.

#### 4.14. Critérios de saída das variáveis

Enquanto as respostas estão sendo processadas no algoritmo, existem alguns critérios de parada que indicam que aquelas respostas já concluíram o seu processamento.

O primeiro critério é a detecção do estado estacionário do valor de  $Q^2$ , cuja técnica está descrita na seção 2.8 deste trabalho. O algoritmo coleta os valores de  $Q^2$  nas últimas “nss” iterações e armazena em um vetor “w”. Em seguida, coleta o número de pontos acumulados em cada iteração das últimas “nss” iterações, armazena em um vetor “x\_”, e realiza uma expansão desse vetor. Então, faz uma interpolação entre “x\_” e “w”. Adiciona um ruído (sugerido 0,3% do sinal de amplitude) e aplica a função “CalculateR\_N” para detectar o estado estacionário.

Esta interpolação é realizada utilizando, por exemplo, um método de interpolação do tipo “spline”. O aumento dos dados de “nss” é necessário para que o algoritmo de detecção de estado estacionário funcione corretamente.

O segundo critério tem relação com o primeiro (verificação do estado estacionário). Uma vez que o estado estacionário for detectado, sabe-se que seus valores pontuais apresentam pequenas variações. Então o algoritmo faz uma interpolação linear para verificar qual é esse valor médio de  $Q^2$  que representa o estado estacionário. Para que o resultado seja satisfatório, este valor deve estar acima do valor mínimo ideal, definido pelo usuário. O valor padrão para  $Q^2$  mínimo é 0,97.

O terceiro critério estabelece que a variável resposta deixará de ser processada quando o valor de  $Q^2$  ficar acima do mínimo permitido ( $lbQ^2$ ), por “nss” iterações consecutivas.

O algoritmo é mais rigoroso nos critérios de saída para as respostas do conjunto “ativo”, diferente da resposta ativa “kresp”, e para o conjunto “não convergido”, porque elas não estão sujeitas às mesmas condições de processamento que a resposta “kresp”, que são: otimização do erro sob as variáveis de decisão como: Tipo de transformação da resposta, método de regressão/função de covariância e método de escalonamento de matrizes.

Para respostas diferentes da resposta “kresp” ativa, são exigidos dois critérios de saída para serem consideradas convergidas, que podem ser: critérios 1 e 2 ou critérios 1 e 3. Para que a resposta “kresp” seja convergida, é exigido o

critério de saída 1 ou 3. Para as respostas do conjunto “não convergido”, os critérios de saída são os mesmos usados para respostas ativas.

Após conferir os critérios de saída, os conjuntos “ativo”, “convergado” e “não convergado” são atualizados.

Se a variável resposta “kresp” atingir o número máximo de amostras e não tiver alcançado o critério de saída, esta variável sairá do conjunto “ativo” e passará para o conjunto “não convergado”.

Após a variável “kresp” ser processada, podendo ter convergado ou não, o algoritmo escolhe uma nova variável ativa “kresp”, o procedimento de escolha é o mesmo descrito na seção 4.10 desta tese. O algoritmo funciona até que não exista mais variáveis no conjunto “ativo”.

O algoritmo armazena e apresenta as informações mais relevantes. Os dados armazenados contêm informações sobre as variáveis regressoras, as manipulações das variáveis respostas, a seleção dos regressores adequados, o metamodelo final e os dados de validação cruzada.

## 4.15. Funções utilizadas no algoritmo

### 4.15.1. VarTransModel

Essa função recebe argumentos e retorna um metamodelo para a resposta que está sendo calculada, em conjunto com o “Erro” com validação cruzada e o valor de  $Q^2$  também com validação cruzada. Os argumentos são um vetor que encapsula índices para o tipo de transformação de resposta, método de regressão/covariância, método de escalonamento, dados de regressão contendo entradas e uma resposta, além dos parâmetros.

A seguir, a função “varTransModel” é apresentada detalhadamente.

---

**Função** VarTransModel

#### **Início**

**Entrada** vTrans //Transformação das respostas, método ou função de covariância, índices de método de escalonamento

regData //dados brutos de regressão

kfold //Número de grupos da validação cruzada

---



---

```

    parX //Parâmetros usados nas funções de
transformações das variáveis regressoras
    parY //Parâmetros usados nas funções de
transformações das variáveis respostas
    par //parâmetros

```

**Saída** `metamodelRegression` //estrutura do metamodelo com os seguintes campos

```

    FlafPredictorTransformation //Sinalizador
verdadeiro/falso indicando se as entradas são
transformadas

```

```

    PredictorExpansionType //Tipo de expansão
das variáveis de entrada: linear, interaction, quadratic,
pure quadratic...

```

```

    PredictorRemovalOnde //Índices de colunas
extraídas da matriz de projeto do preditor

```

```

    PredictorScaling1 //Fator de pré-
escalamento

```

```

    PredictorScaling2 //Fator de pré
escalamento

```

```

    PredictorColumnRemoval //Índices das
colunas restantes da matriz de projeto do preditor após a
remoção de colunas duplicadas dentro da tolerância

```

```

    PredictorRankAdjustment //Índices das
colunas restantes da matriz de projeto do preditor após o
ajuste de classificação

```

```

    SelectedPredictors //vetor de índices
preditores selecionados pela seleção de recursos baseada
em mínimos quadrados ordinários (OLS) com validação cruzada

```

```

    ResponseTransformationIndex //Índice das
transformações das variáveis respostas

```

```

    ResponseScaling1 //Fator de pré-
escalamento das variáveis respostas

```

```

    ResponseScaling2 // Fator de pré-
escalamento das variáveis respostas

```

```

    ResponseData //Dados das variáveis
respostas transformadas e escalonadas

```

```

    Metamodel //Metamodelo selecionada
SCOREcv //Valor do Erro da validação
cruzada

```

```

    Q2cv //Valor do  $Q^2$  da validação cruzada

```

```

    RegressionType //Descrição do método de
regressão resultante

```

```

    ResponseRange //Imagem das variáveis
respostas

```

```

    MethodType //Método dos mínimos quadrados
ou função de covariância

```

```

    validationErro //Métrica valor do Erro_r da
validação cruzada

```

```

    validationQ2 //Valor do  $Q^2$  da validação cruzada

```

**Função** `funVarTrans`

---

---

**Retorne** Resposta única transformada pela função de transformação elementar

**Fimfunção**

**Função** standNorm

**Retorne** //Variável resposta escalonada com o mesmo parâmetro de escala usado pelas entradas

**Fimfunção**

**Função** licols

//Transformar, expandir, remover colunas de unidades, dimensionar, remover colunas duplicadas e ajustar a classificação da matriz de projeto do

**Fimfunção**

**Selecione** //Regressores usando seleção de recursos baseada em mínimos quadrados ordinários com validação cruzada

**Função** mysequentialFeatureSelection

**Retorne** //regressores selecionados nas duas direções (menor Erro)

**Armazene** sel

**Fimfunção**

**Atribua** Erro final com validação cruzada

**Calcule** Q2 com validação cruzada

**Armazene** //Metamodelo de regressão linear pelo método dos mínimos quadrados

//Caso seja atribuída a regressão não linear, deve-se seguir as seguintes etapas.

**Se** método de regressão é a regressão de processo gaussiana (GPR) usando a caixa de ferramenta de processos gaussianos para aprendizagem de máquina (GPML) **Então**

**Leia** maxEval, inffunc, meanfunc, likfun

**Atribua** //Função de covariância designada

**Leia** //Valores iniciais dos hiperparâmetros

**Otimize** //hiperparâmetros

//Construa o metamodelo (GPR) correspondente

**Faça** //validação cruzada do metamodelo resultante

**Selecione** //metamodelo com o menor Erro entre o modelo de regressão linear e não linear, ambos com validação cruzada.

**Armazene** //metamodelo final de regressão

**Fimse**

**Fimfunção**

---

A função “varTransModel” é responsável pelo tratamento das variáveis regressoras, tais tratamentos são: Modificação das entradas que são de baixa dimensão, passando-as para alta dimensão através de funções de transformações elementares; Expansão das entradas transformadas resultantes em termos lineares, de interação, quadráticos e quadráticos puros; Remoção de

colunas de termos de interação durante a expansão e escalonamento; Remoção de colunas duplicadas que possam aparecer; Realização de ajuste de classificação removendo colunas linearmente dependentes; Transformação da variável resposta devido à aplicação da função de transformação elementar designada.

Os regressores resultantes e a resposta transformada são, então, sujeitos à validação cruzada, tanto nas direções para frente quanto para trás, a partir da qual a direção com o menor erro quadrático (Erro) é escolhida e, em seguida, o Erro e  $Q^2$  validados de forma cruzada são calculados juntamente com o modelo de regressão dos mínimos quadrados correspondente.

Se for atribuída a regressão não linear, a função utilizará os regressores já selecionados do procedimento de seleção baseado em mínimos quadrados e a função de covariância para calcular o Erro por validação cruzada. Com o cálculo feito, a função irá comparar o erro obtido pela regressão não linear com o erro do modelo de regressão pelo método dos mínimos quadrados. Então, o menor erro determinará o metamodelo de escolha.

Para a seleção dos regressores baseado no método dos mínimos quadrados com validação cruzada, é necessário definir o critério usado para a validação. O cálculo do erro é apresentado na equação a seguir:

$$Erro(X_{tr}, y_{tr}, X_{te}, y_{te}) = \frac{\|y_{te} - X_{te}^* [(X_{tr}^{*T} X_{tr}^*)^{-1} X_{tr}^{*T} y_{tr}]\|_2^2}{\|y_{te} - \bar{y}_{te}\|_2^2} \quad (222)$$

Onde  $X_{tr}$  é a matriz de projeto dos regressores para treinamento;  $y_{tr}$  é o vetor da resposta transformada para treinamento;  $X_{te}$  é a matriz de projeto dos regressores para teste;  $y_{te}$  é o vetor da resposta transformada para teste;  $X_{tr}^*$  e  $X_{te}^*$  são, respectivamente, as matrizes de projeto de regressores estendidos para treinamento e teste, com uma coluna inicial de uns;  $\bar{y}_{te}$  é a média aritmética de  $y_{te}$ ;  $(X_{tr}^{*T} X_{tr}^*)^{-1} X_{tr}^{*T} y_{tr}$  é o vetor dos coeficientes da regressão linear.

O termo  $X_{te}^* [(X_{tr}^{*T} X_{tr}^*)^{-1} X_{tr}^{*T} y_{tr}]$ , presente na equação anterior representa a resposta prevista para os dados de teste usando o metamodelo linear que foi regredido com dados de treinamento.

A função “mysequentialFeatureSelection” é uma modificação do “sequentialfs” do Matlab, onde é possível incluir/excluir grupos de variáveis e acelerar o processo de seleção.

Com o cálculo do Erro final e do  $Q^2$  com validação cruzada, a função armazena o metamodelo de regressão linear dos mínimos quadrados. Este modelo insere o argumento dos regressores manipulados dentro da função e retorna a resposta manipulada. Então a função “mypredict” é acionada para retornar a variável resposta à forma inicial (antes da transformação).

Caso seja atribuída a regressão não linear, o algoritmo usará a toolbox GPML – Processos Gaussianos para aprendizagem de máquina onde estará as vinte e cinco funções de covariância. A toolbox já foi descrita na seção 2.9.

#### 4.15.2. “Predictors”

A seguir, a função “Predictors” é apresentada detalhadamente. Esta função, conforme explicada anteriormente, recebe argumentos e transforma as entradas (baixa dimensão) em regressores (alta dimensão) aplicando funções de transformações elementares; Expande as entradas transformadas resultantes em termos lineares, de interação, quadráticos e quadráticos puros; Remove colunas de termos de interação durante a expansão; Remove colunas duplicadas; Realizam ajustes de classificação removendo colunas linearmente dependentes (função “licols”).

---

**Função** Predictors

**Início**

**Entradas**

```

X-m-by-ni //matriz de dados de entrada antes de
qualquer manipulação
fT //Escolha de transformação dos regressores
(verdadeiro/falso)
fE //Tipo de expansão dos regressores (linear,
interaction, quadratic, purê quadratic ou outro definido
pelo usuário
fR // Escolha de ajuste de classificação
(verdadeiro ou falso)
parX //Conjunto de parâmetros das transformações
dos regressores

```

---

---

```

        flagScl //Escala de entrada (0 - nenhuma; 1 -
padronização; 2 - normalização; 3 - normalização
euclidiana)
        nfun //número de funções de transformações dos
regressores
        par //parâmetros
Saídas
        xTrans //dados das variáveis regressoras
transformadas, expandidas, escalonadas, removidas de
colunas e ajustadas por classificação
        i0 //Remoção do regressor da coluna da matriz d
índice "uns"
        S1x //matriz de fator do pré escalonamento
        S2x //matriz de fator do pós escalonamento
        iA //matriz de índice de remoção de coluna dos
regressores
        iK //matriz de índice de ajuste de classificação
dos regressores

Se fT ← verdadeiro Então
        Função funVarTrans
                Retorne //Variáveis regressoras
transformadas
        Fimfunção
Senão
        XTrans ← X
Fimse
Função x2fx
        Retorne //variáveis regressoras transformadas de
acordo com a especificação em fE (linear, interaction,
quadratic, purê quadratic ou outra especificada pelo
usuário)
        Fimfunção
Leia toLo //tolerância para coluna com valores "uns"
//Remoção das colunas de "uns", com tolerância, que
pode aparecer devido à etapa de expansão dos regressores
Armazene i0 //os índices das colunas que restaram após
a remoção dos "uns"
Função standNorm
        Retorne Matriz dos regressores escalonada de
acordo com o método escolhido em "flagScl"
        Armazene S1x, S2x //fatores de pré escalonamento
da matriz dos regressores
Fimfunção
Função uniqueTolColumns
        Retorne Matriz resultante da etapa anterior com
a remoção das colunas duplicadas com tolerância de 10-12
        Armazene iA //Vetor de índice das colunas
restantes em ordem crescente
Fimfunção
Se fR ← verdadeiro Então

```

---

---

```

Se xTrans tem mais colunas do que linhas Então
    //Mantenha todos os regressores
    Armazene xTrans //matriz final dos
regressores da etapa anterior
    Senão
        Se a classificação de xTrans da etapa
anterior está cheia Então
            //Mantenha todos os regressores
            Armazene xTrans //matriz final dos
regressores da etapa anterior
        Senão
            Função licols
                Retorne Matriz com as variáveis
regressoras excluindo as variáveis dependentes
            Fimfunção
            Armazene iK //Colunas restantes da
etapa anterior
            Armazene xTrans //Matriz final com os
regressores contendo as colunas definidas no vetor iK
        Fimse
    Fimse
Senão
    //Mantenha todos os regressores
    Armazene xTrans //matriz final dos regressores
da etapa anterior
    Fimse
Fimfunção

```

---

A utilização da função “funVarTrans” pela função “Predictors” transformará as variáveis regressoras da seguinte forma: A primeira coluna da matriz transformada conterá os regressores transformados pela primeira função de transformação, a segunda coluna conterá os regressores transformados pela segunda função de transformação, e assim por diante.

Na etapa de remoção de “uns”, a função verifica se pelo menos um elemento de cada coluna de  $|xTrans - ones(size(xTrans))|$  é menor do que a tolerância “toLo”, e então faz a remoção da coluna. As colunas restantes são armazenadas na variável “iO”.

### 4.15.3. fscoreGpml

A seguir, a função “fscoreGpml” é apresentada detalhadamente. Esta função pega o argumento e retorna valores para a validação cruzada ao usar o “crossval” do Matlab com a caixa de ferramentas GPML.

---

**Function** fscoreGpml

**Início**

**Entrada**

$X_{tr-m_{tr}-by-n_p}$  //transforma a matriz de design de treinamento,  $m_{tr}$  representa o tamanho do conjunto de treinamento como definido pelo kfold e  $n_p$  é representa o número de regressores selecionados

$Y_{tr-m_{tr}-by-1}$  //vetor da respsta transformada para treinamento

$X_{te-m_{tr}-bt-n_p}$  //matriz dos regressores de projeto transformada para teste, tamanho do conjunto de treinamento conforme definido por kfold

$Y_{te-m_{te}-by-n_p}$  //vetor da resposta transformada para teste

parGpml //conjunto de parâmetros do metamodelo GPML

**Saída**

Scores //matriz de pontuações usadas para validação cruzada [MSR,  $Q^2$ , SER]

**Colete** //Funções de inferência, média, covariância e verossimilhança de acordo com a forma das funções de média e covariância definidas em "varTransModel"

**Colete** //Valores dos hiperparâmetros

**Calcule** //a resposta prevista para o metamodelo de regressão de processo gaussiano treinado usando a função "gp" do GPML

$see \leftarrow \|y_{te} - \hat{y}\|_2^2$

$sst \leftarrow \|y_{te} - \bar{y}_{te}\|_2^2$

$MSE \leftarrow see/m_{te}$

$RSE \leftarrow see/sst$

$Q^2 \leftarrow 1-RSE$

**Colete** MSE,  $Q^2$ , SER

---

**Fimfunção**

#### 4.15.4. varTransModelOptimization

Essa função recebe argumentos, executa otimizações de inteiros e retorna variáveis usadas pelo algoritmo de chamada. As variáveis de decisão para a otimização são a função de transformação de resposta, o método de regressão/função de covariância e os índices do método de escalonamento. A seguir, a função "varTransModelOptimization" é apresentada detalhadamente.

---

**Function** varTransModelOptimization

---

---

## Início

### Entrada

```

regData //lista com dados brutos de regressão
kfold //Número de grupos na validação cruzada
parX //parâmetros usados pelas funções de
transformações das variáveis regressoras
parY //parâmetros usados pelas funções de
transformações das variáveis respostas
lbQ2 //limite inferior do valor de  $Q^2$ 
Mmax //Número máximo de avaliações de função para
cada reinicialização de otimização (Mmax = 30)
optMethod //Método utilizado para otimizar a
resposta "kresp".
par //parâmetros

```

### Saída

```

mdl //metamodelo selecionado da otimização
vTrans //lista com os valores das variáveis de
decisão a partir da otimização: transformação das
respostas, método de regressão/função de covariância e
índices dos métodos de escalonamento
score //métrica do erro, RSE
Q2 //Valor do  $Q^2$  da validação cruzada

```

**Colete** nfunY, nreg, nscale //número de funções de transformações da variável resposta para a resposta ativa "kresp"; número de modelos de regressão; número dos métodos de escalonamento.

**Verifique** optMethod

**Caso** 1 //Abordagem de força bruta

**Selecione** //melhor metamodelo com o menor "score" entre todas as possíveis permutações de funções de transformações das respostas, método/função de covariância e métodos de escalonamento.

**Use** //gráficos para checar o progresso

**Caso** 2 //Otimização substituta

**Defina** //RSE com validação cruzada como a função objetivo a ser minimizada "varTransObj"

**Leia** ub, lb //limites inferiores e superiores para as variáveis de decisão (lb=[1,1,1] e ub=[nfunY,nreg,nscale])

**Defina** //Todas as variáveis de decisão como variáveis inteiras para otimização

**Leia** nrestarts //número de restarts da otimização (nrestarts=5)

**Calcule** //número máximo de avaliações de funções em cada restart

**Processe** surrogateopt //rotina de otimização

---



---

```

Use //gráficos para analisar o progresso
Armazene vTrans, score
Selecione //função de transformação da
resposta em "par.funkresp" baseado no índice em "vTrans"
Função varTransModel
Retorne mdl, Q2
Fimfunção
Caso 3
Defina //Erro da validação cruzada como
função objetivo a ser minimizada
Defina zT, zS (especifique variáveis,
limites e tipo inteiro)
Leia nrestarts (nrestarts = 5)
Calcule //número máximo de avaliações de
funções em cada restart
Processe surrogateopt //rotina de
otimização

Use //gráficos para analisar o progresso
Armazene vTrans, score
Selecione //função de transformação da
resposta em "par.funkresp" baseado no índice em "vTrans"
Função varTransModel
Retorne mdl, Q2
Fimfunção
Case 4
Defina //Erro da validação cruzada como
função objetivo a ser minimizada
Defina zT, zS, zM (especifique variáveis,
limites e tipo inteiro)
Leia nrestarts (nrestarts = 5)
Calcule //número máximo de avaliações de
funções em cada restart
Processe surrogateopt //rotina de
otimização

Use //gráficos para analisar o progresso
Armazene vTrans, score
Selecione //função de transformação da
resposta em "par.funkresp" baseado no índice em "vTrans"
Função varTransModel
Retorne mdl, Q2
Fimfunção
Fimverifique

```

---

### **Fimfunção**

Para a resposta "kresp", o algoritmo utiliza um método de otimização para encontrar o melhor metamodelo. O usuário pode escolher qual é o método de otimização que será utilizado: 1 – testar todas as combinações possíveis (não recomendado); 2 – Otimização substituta; 3 ou 4 – Otimização Bayesiana.

Para o caso 3 é utilizado um modelo de otimização bayesiana com duas variáveis de decisão: função de transformação das respostas e método de escalonamento. Esta otimização é indicada quando o algoritmo usará apenas o modelo de regressão linear para a resolução do problema.

Para o caso 4 é utilizado um modelo de otimização bayesiana com três variáveis de decisão: função de transformação da resposta, método de regressão/função de covariância e método de escalonamento. Indicado quando o modelo

O número máximo de avaliações de funções em cada restart é dado pela equação a seguir.

$$M = \min \left\{ \left\lceil \frac{nfunY.nreg.nscale}{3 \cdot restarts} \right\rceil, Mmax \right\} \quad (223)$$

Heuristicamente, é requerido que o número máximo de avaliações de funções seja 1/3 do número total de possibilidades. Assim, o resultado é dividido em “restarts” para se chegar ao número máximo de avaliações de funções. Entretanto, é limitado  $M$  ao  $Mmax$  de tal forma que o número total de avaliações de funções é  $M \times restart \leq Mmax \times 5 = 30 \times 5 = 150$ . Esta é uma afirmação heurística para encontrar um mínimo “bom” para o objetivo da otimização em um tempo de processamento razoável.

#### 4.15.5. varTransObj

Essa função recebe argumentos e retorna o valor da função objetivo a ser minimizada pela função “varTransModelOptimization” A seguir, a função “varTransObj” é apresentada detalhadamente.

---

**Função** varTransObj

**Início**

**Entrada**

VTrans //matriz ou tabela (depende da rotina de otimização usada) contendo índice da função de transformação da resposta, método de regressão/função de covariância, índice do método de escalonamento  
regData //matriz com dados brutos de regressão

---

---

```

    kfold //número de conjuntos para validação
cruzada
    parX //parâmetros usados pelas funções de
transformações das variáveis regressoras
    parY //parâmetros usados pelas funções de
transformações das variáveis respostas
    flag //Tipo de otimização (surrogate,
bayesian2vars, bayesian3vars)
    par //parâmetros

```

**Saída**

```
score //métrica de erro (Erro relativo RSE)
```

**Verifique** flag

```
Caso surrogate //varTransModelOptimization
utiliza otimização substituta
```

```
Selecione vTrans(1) //função de
transformação da variável resposta
```

```
Função varTransModel
```

```
Retorne score
```

**Fimfunção**

```
Caso bayesian2vars //varTransModelOptimization
utiliza otimização bayesiana
```

```
Converta vTrans //de tabela para Matriz
```

```
Selecione vTrans(1) //função de
transformação da variável resposta
```

```
Ajuste //matriz resultante de acordo com o
argumento adequado da função "varTransModel"
```

```
Função varTransModel
```

```
Retorne score
```

**Fimfunção**

```
Caso bayesian3vars //varTransModelOptimization
utiliza otimização bayesiana
```

```
Converta vTrans //de tabela para Matriz
```

```
Selecione vTrans(1) //função de
transformação da variável resposta
```

```
Função varTransModel
```

```
Retorne score
```

**Fimfunção****Fimverifique**

```
Fimfunção
```

---

**4.15.6. funVarTrans**

Essa função recebe argumentos e transforma os dados da matriz, seja das variáveis regressoras ou das variáveis respostas, usando funções elementares definidas em “parameters”, sendo uma função elementar por vez. A seguir, a função “funVarTrans” é apresentada detalhadamente.

---

**Função** funVarTrans

**Início**

**Entrada**

```

zTrans //Vetor com o índice da função de
transformação elementar para cada variável
Z //matriz de dados a serem transformados
(regressores e respostas). Pode ser uma matriz
(regressores) ou um vetor (respostas)
parScale //Matriz dos parâmetros usados pelas
funções de transformações: “parX” para regressores; “parY”
para respostas.
varType //regressor, resposta ou o cálculo da
resposta inversa (predictor, response, inverse)
par //parâmetros

```

**Saída**

```

y //dados da matriz transformada

```

**Verifique** varType

```

Caso predictor //transformar variáveis
regressoras

```

```

Use //função definida em parâmetros
“par.fun” indexada por “zTrans” com argumentos “Z” e
“parScale.

```

```

Armazene y //resultado da transformação

```

```

Caso response //transformar variáveis respostas

```

```

Use //função definida em parâmetros
“par.fun” indexada por “zTrans” com argumentos “Z” e
“parScale.

```

```

Armazene y //resultado da transformação

```

```

Caso inverse //usado para inverter a variável
resposta, voltar a sua forma original

```

```

Use //função definida em parâmetros
“par.fun” indexada por “zTrans” com argumentos “Z” e
“parScale.

```

```

Armazene y //resultado da transformação

```

**Fimverifique**

**Fimfunção**

---

#### 4.15.7. simulFuncAspenPlus

Essa função recebe argumentos e simula o modelo específico no Aspen Plus, sendo uma linha de valor de entrada por vez. A seguir, a função “simulFuncAspenPlus” é apresentada detalhadamente.

---

**Função** simulFuncAspenPlus

**Início**

**Entrada**

```

points //matriz com as entradas (variáveis
independentes) para ser enviada ao simulador
in_ //Vetor de objeto das variáveis de entrada
especificadas tal qual Variable Explorer do Aspen Plus
out_ //Vetor de objeto das variáveis de saída
especificadas tal qual Variable Explorer do Aspen Plus
aspen //Servidor do Aspen Plus "ActiveX"
especificando a simulação

```

**Saída**

```

values //valores de respostas após as simulações
no Aspen Plus
convFlag //indicador de convergência da
simulação (8 - convergido; 9 - não convergido; 10 -
atenção)

```

```

Atribua //valores de entrada em "points" ao objeto de
entrada "in_"

```

```

Simule Aspen plus especificado em "aspen"

```

```

Armazene convflag, values

```

**Fimfunção**

---

#### 4.15.8. simulFuncSimulink

Essa função recebe argumentos e simula o modelo específico no Simulink, sendo uma linha de valor de entrada por vez. A seguir, a função “simulFuncSimulink” é apresentada detalhadamente.

---

**Função** simulFuncSimulink

**Início**

**Entrada**

```

points //matriz com as entradas (variáveis
independentes) para ser enviada ao simulador
x //Valores de variáveis no estado nominal usados
como parâmetros no Simulink
sim time //tempo de simulação

```

---

---

```
optionsSimulink //opções de simulação dinâmica
```

**Saída**

```
values //valores de respostas após as simulações
no Simulink
```

```
Atribua //valores de entrada em "points" ao objeto de
entrada "in_"
```

```
Simule modelo em Simulink com as novas variáveis de
entrada
```

```
Armazene values
```

```
Fimfunção
```

---

**4.15.9. simulFuncMatlab**

Essa função recebe argumentos e executa a função Matlab especificada onde os dados de respostas são gerados, sendo uma linha de valores de entrada por vez. A seguir, a função "simulFuncMatlab" é apresentada detalhadamente.

---

```
Função simulFuncMatlab
```

**Início****Entrada**

```
points //matriz com as entradas (variáveis
independentes) para ser enviada ao simulador
```

**Saída**

```
values //valores de respostas após as simulações
```

```
Conecte //valores da variável de entrada em cada uma
das funções do banco de testes
```

```
Armazene values
```

---

As funções do banco de testes são equações matemáticas de difícil resolução e serão apresentadas a seguir, conforme mostradas por Surjanovic e Bingham (2013). A última função foi criada pela própria autora deste trabalho e envolve muitas funções matemáticas, dificultando a sua resolução.

**4.15.9.1. Função de Rastrigin**

Esta função tem vários mínimos locais regularmente distribuídos, conforme mostrada na Figura 2.

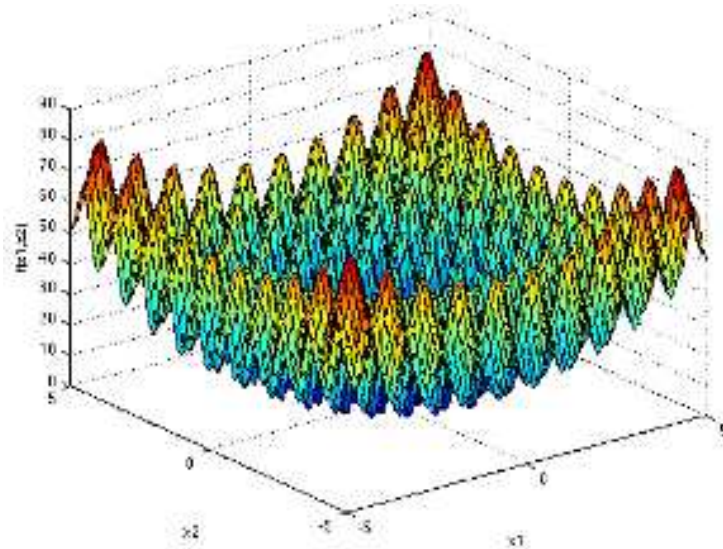


Figura 3: Função de Rastrigin. Fonte: Surjanovic e Bingham (2013)

A equação para a função de Rastrigin é apresentada a seguir.

$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)] \quad (224)$$

#### 4.15.9.2. Função de Schwefel

É uma função complexa com muitos mínimos locais, conforme mostrada na Figura 3.

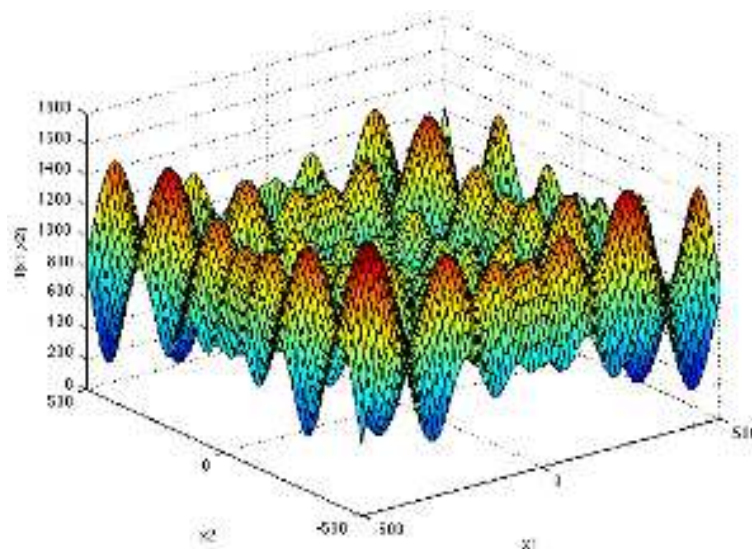


Figura 4: Função de Schwefel. Fonte: Surjanovic e Bingham (2013)

A equação para a função de Schwefel é apresentada a seguir.

$$f(x) = 418,9829d - \sum_{i=1}^d x_i \text{sen}(\sqrt{|x_i|}) \quad (225)$$

#### 4.15.9.3. Função hiperelipsoide rotacionada

Trata-se de uma função contínua, convexa e unimodal e é mostrada na Figura 4.

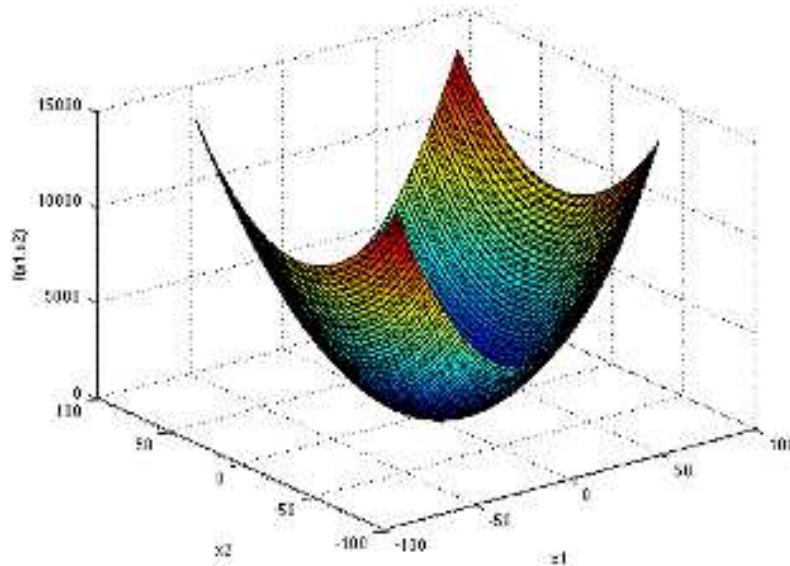


Figura 5: Função hiperelipsoide rotacionada.  
Fonte: Surjanovic e Bingham (2013).

A equação para a função hiperelipsoide rotacionada é apresentada a seguir.

$$f(x) = \sum_{i=1}^d \sum_{j=1}^i x_j^2 \quad (226)$$

#### 4.15.9.4. Função de Styblinski-Tang

Esta função é apresentada na Figura 5 em sua forma bidimensional.

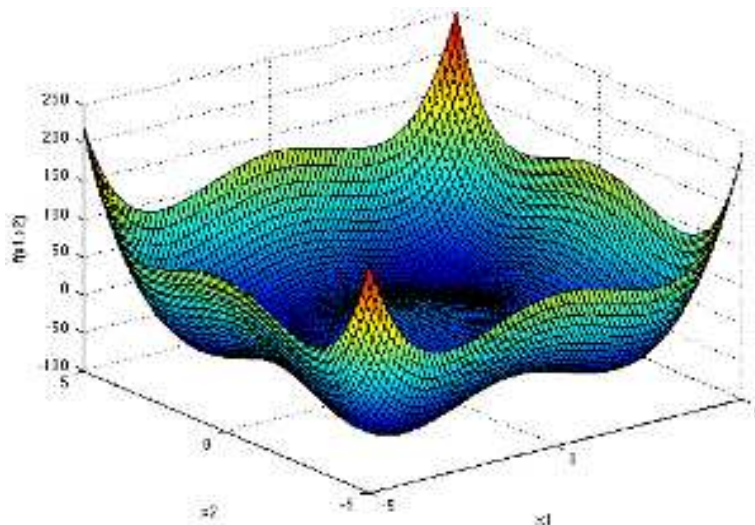


Figura 6: Função de Styblinski-Tang.  
Fonte: Surjanovic e Bingham (2013).



A equação para a função de Styblinski-Tang é apresentada a seguir.

$$f(x) = \frac{1}{2} \sum_{i=1}^d (x_i^4 - 16x_i^2 + 5x_i) \quad (227)$$

#### 4.15.9.5. Função de Zharakov

Esta função não apresenta mínimos locais, apenas o mínimo global e é apresentada na Figura 6 em sua forma bidimensional.

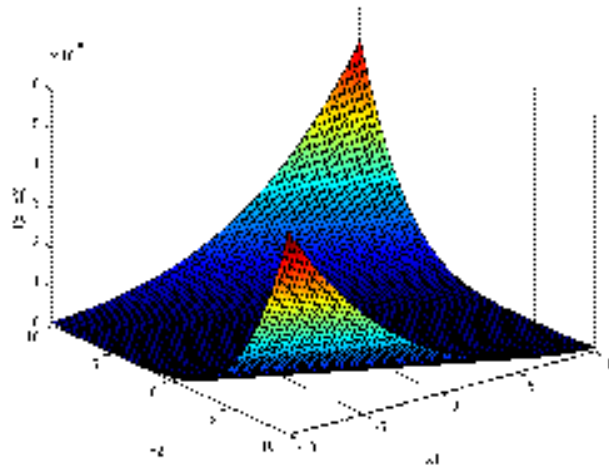


Figura 7: Função de Zhakarov.  
Fonte: Surjanovic e Bingham (2013).

A equação para a função de Zakharov é apresentada a seguir.

$$f(x) = \sum_{i=1}^d x_i^2 + (\sum_{i=1}^d 0,5ix_i)^2 + (\sum_{i=1}^d 0,5ix_i)^4 \quad (228)$$

#### 4.15.9.6. Função da soma dos quadrados

Esta é uma função contínua, convexa e unimodal e possui apenas mínimo global. Sua forma bidimensional é apresentada na Figura 7.

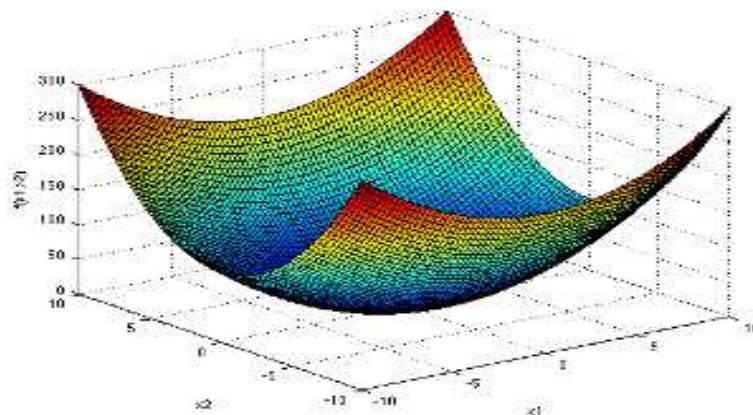


Figura 8: Função da soma dos quadrados.  
Fonte: Surjanovic e Bingham (2013).

A equação para a função de Zakharov é apresentada a seguir.

$$f(x) = \sum_{i=1}^d ix_i^2 \quad (229)$$

#### 4.15.9.7. Função de Ackley

Função bastante utilizada como teste em algoritmos de otimização. Sua forma bidimensional é apresentada na Figura 9 e é caracterizada por uma região externa quase plana e um grande buraco no centro.

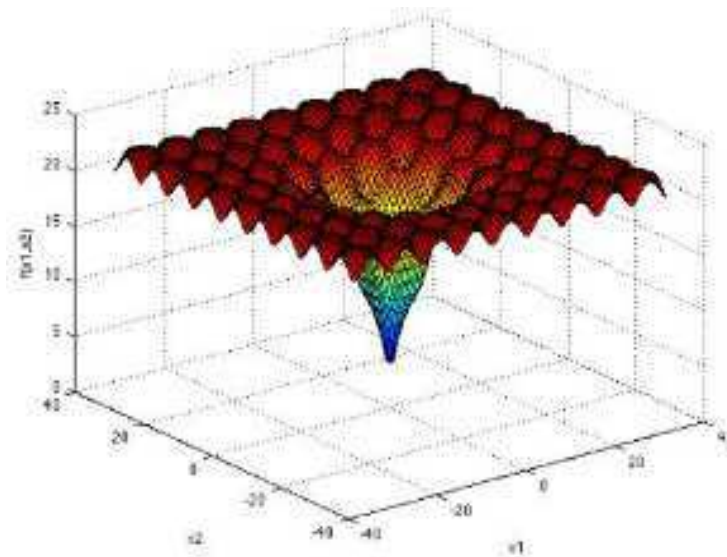


Figura 9: Função de Ackley.  
Fonte: Surjanovic e Bingham (2013).

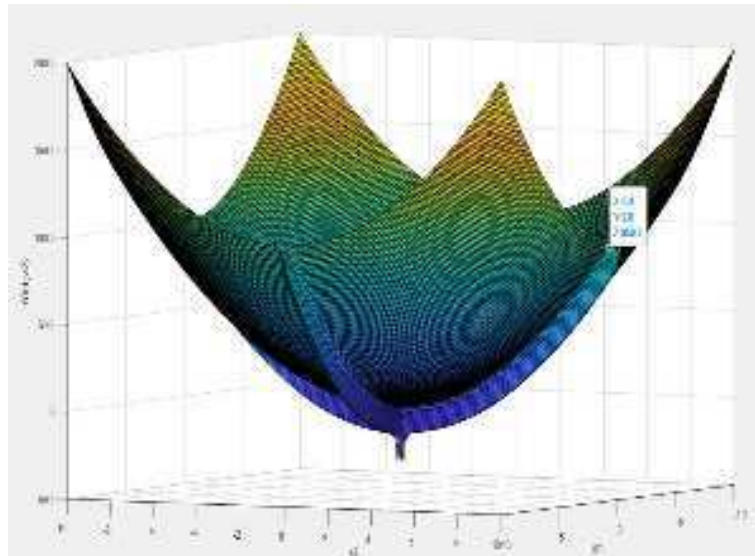
A equação para a função de Ackley é apresentada a seguir.

$$f(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i)\right) + a + \exp(1) \quad (230)$$

Onde os valores recomendados para as constantes são  $a = 20$ ,  $b = 0,2$  e  $c = 2\pi$ .

#### 4.15.9.8. Função autoral

Esta função possui muitos termos matemáticos, o que dificulta o seu processo de resolução. A Figura 10 mostra o gráfico da função.



**Figura 10: Função Autoral**  
**Fonte: Próprio Autor**

A equação que representa esta função é apresentada a seguir.

$$f(x) = \sum_{i=1}^d \left( 10x_i^2 + \frac{5}{2} \cos(x_i) - e^{\frac{x_i}{10}} + \pi \operatorname{sen}\left(\frac{1}{x_i}\right) - \frac{1,6}{x_i^2} \right) \quad (231)$$

#### 4.15.10. standNorm

Essa função recebe argumentos e executa o escalonamento do seu argumento de entrada. O tipo de escalonamento pode ser: sem escalonamento, normalização, normalização euclidiana ou outro tipo de escalonamento definido pelo usuário. A fórmula geral do escalonamento é dada por:

$$X_S = \frac{X - S_1}{S_2} \quad (232)$$

As variáveis serão explicitadas detalhadamente na função “standNorm”, mostrada a seguir.

---

**Função** standNorm

**Início**

**Entrada**

```
X //matriz antes do escalonamento (uma linha com
os valores de entrada)
flagScl //escolha do método de escalonamento (0
- nenhum; 1 - padronização; 2 - normalização; 3 -
normalização euclidiana;
```

---

---

```

    S1 //parâmetro de pré escalonamento (usado
quando flagScl não é especificado)
    S2 //parâmetro de pós escalonamento (usado
quando flagScl não é especificado)

```

**Saída**

```

    Xs //Matriz após escalonamento
    S1 //parâmetro de pré escalonamento (usado
quando flagScl não é especificado)
    S2 //parâmetro de pós escalonamento (usado
quando flagScl não é especificado)

```

**Se flagScl está especificado Então****Verifique** flagScl

```

    Caso 0 //Xs=X
        S1 ← 0 //Vetor de "zeros" na mesma
quantidade de variáveis em X
        S2 ← 1 //Vetor de "uns" na mesma
quantidade de variáveis em X
        Xs ← X

```

```

    Caso 1 //padronização  $Xs = (X - \bar{X})/\sigma_x$ 

```

```

        S1 ← "média de X"
        S2 ← "desvio padrão de X"
        Xs ← (X-S1)/S2

```

```

    Caso 2 //Normalização  $Xs = (X - Xmin)/(Xmax -$ 
Xmin)

```

```

        S1 ← "mínimo de X"
        S2 ← "máximo de X"
        S2 ← S2-S1
        Xs ← (X-S1)/S2

```

```

    Caso 3 //normalização euclidiana  $Xs = (X -$ 
 $\bar{X})/\|X - \bar{X}\|_2$ 

```

```

        S1 ← "média de X"
        S2 ←  $\|X-S1\|_2$ 
        Xs ← (X-S1)/S2

```

**Fimverifique**

```

Senão //Parâmetros S1 e S2 definidos pelo usuário

```

```

    Xs ← (X-S1)/S2

```

**Fimse****Fimfunção****4.15.11. checkpoint**

Essa função recebe os argumentos, detecta e substitui os "zeros" por valores adequados na matriz de projeto. Esses valores iguais a "zero" podem surgir na matriz de projeto durante a geração de amostras pelo método de amostragem sequencial. A seguir, a função "checkpoint" é apresentada detalhadamente.

---

**Função** checkpoint

**Início**

**Entrada**

```

        points //matriz de projeto das variáveis de
entrada
        xmin //limites inferiores das variáveis de
entrada
        xmax //limites superiores das variáveis de
entrada

```

**Saída**

```

        points //matriz de projeto da variáveis de
entrada após a substituição dos "zeros" por valores
adequados

```

```

Identifique //índices lógicos dos elementos de xmax
que contém os valores "zero"

```

**Armazene** e

```

Identifique //índices das colunas que contenha os
valores "zeros"

```

**Armazene** col

```

dz ← (-1)^e(col) . [0,0001(xmax(col)-xmin(col))

```

```

Substitua //valores "zeros" por dz

```

**Fimfunção**

---

#### 4.15.12. myPredict

Esta função recebe argumentos e prevê respostas para um metamodelo específico. Essa função manipula a matriz dos dados de entrada por meio de transformação, expansão, escalonamento, remoção de coluna e ajuste de classificação para tornar a resposta prevista em escalonada e não escalonada. A seguir, a função "myPredict" é apresentada detalhadamente.

---

**Função** myPredict

**Início**

**Entrada**

```

        X //Matriz de dados de entrada antes de qualquer
manipulação
        y //vetor coluna dos dados de resposta antes da
transformação e escalonamento
        metamodelRegression //estrututa do metamodelo
contendo

```

---

---

```

        FlagPredictorTransformation //indicador
verdadeiro ou falso mostrando quais variáveis de entrada
são transformadas
        PredictorExpansionType //Tipo de expansão
das variáveis respostas - linear, interação, quadrática,
quadrática puro
        PredictorRemovalOne //índice de colunas com
"uns" extraídas da matriz de projeto dos regressores
        PredictorScaling1 //Fator de pré
escalamento dos regressores
        PredictorScaling2 //Fator de pós
escalamento dos regressores
        PredictorRankAdjustment //índices das
colunas restantes da matrix de projeto dos regressores após
ajuste de classificação
        SelectedPredictors //vetor de índices dos
regressores selecionados pelo método dos mínimos quadrados
com validação cruzada
        ResponseTransformationIndex //índices das
transformações das variáveis respostas
        ResponseScaling1 //Fator de pré
escalamento da variável resposta
        ResponseScaling2 //Fator de pós
escalamento da variável resposta
        ResponseData //variáveis respostas
transformadas e escalonadas
        Metamodel //metamodelo selecionado
        SCOREcv //Erro RSE da validação cruzada
        Q2cv //Q2 da validação cruzada
        Coefficients //coeficientes de regressão
dos mínimos quadrados ordinários
        RegressionType //descrição do método de
regressão resultante (mínimos quadrados ou função de
covariância)
        ResponseRange //Imagem das respostas
        MethodType //Mínimos quadrados ou função de
covariância
        parX //parâmetros usados pelas funções de
transformações das variáveis regressoras
        parY // parâmetros usados pelas funções de
transformações das variáveis respostas
        par //parâmetros

```

### Saída

```

        xTrans //Matriz dos regressores após
transformação, expansão, remoção de "uns", escalamento,
remoção de colunas e ajuste de classificação
        zTrans //Vetor de resposta atual transformado e
escalado
        yhats //vetor de resposta previsto transformado
e escalado

```

---

---

yhat //vetor de resposta previsto não escalonado e não transformado

**Se** FlagPredictorTransformation é verdadeiro **Então**

**Função** funVarTrans

**Retorne** xTrans //matriz das variáveis regressoras após aplicação das funções de transformações elementares

**Fimfunção**

**Senão**

xTrans ← X

**Fimse**

**Verifique** PredictorExpansionType

**Função** x2fx

**Retorne** xTrans //matriz das variáveis regressoras após aplicação da técnica de expansão

**Fimfunção**

**Fimverifique**

**Verifique** se tem alguma coluna de "uns" após a etapa de expansão e remova-as

**Função** standNorm

**Retorne** xTrans //matriz das variáveis regressoras após técnica de escalonamento indicada

**Fimfunção**

**Verifique** se tem alguma coluna de "uns" após a etapa de escalonamento e remova-as

**Verifique** se existe duplicação de colunas e remova-as

**Função** funVarTrans

**Retorne** //Variáveis respostas transformadas

**Fimfunção**

**Função** standNorm

**Retorne** //Variáveis respostas escalonadas

**Fimfunção**

**Calcule** yhats //aplicando "metamodel" à matriz das variáveis regressoras final

yhat ← "ResponseScaling1" + "yhats"."ResponseScaling2"

**Função** funVarTrans

**Retorne** //transformação inversa de "yhat"

**Fimfunção**

**Armazene** rangeInvY //imagem da variável resposta inversa

**Se** ResponseRange são reais negativos e rangeInvY é diferente de reais **Então**

yhat ← -yhat

**Fimse**

**Fimfunção**

---

#### 4.15.13. uniqueTolComumns

Esta função remove colunas redundantes da matriz de dados conforme medido por algum parâmetro de norma vetorial dentro de uma determinada tolerância. A função retorna um vetor lógico de índices de forma que a matriz de dados atualizada retenha as colunas exclusivas da matriz de dados original (FLAKKE, 2021).

---

**Function** uniqueTolColumns

**Início**

**Entrada**

X //Matriz de projeto dos regressores  
 tol //tolerância para remoção das colunas  
 p //tipo da norma vetorial

**Saída**

iX //vetor lógico de índices das colunas retidas  
 de X

**Leia** n //número de colunas de X

**Para** todo  $i \leq n$  **faça**

**Se**  $\|X(:, i) - X(:, j)\|_p < tol$  **Então**

$iX \leftarrow false$  //cada coluna de X é checada  
 contra todas as outras colunas de X

**Fimse**

**Fimpara**

**Fimfunção**

---

O tipo da norma vetorial, apresentado na função pode ser de diferentes formas.  $p = 1$ , para:

$$\|v\|_1 = \sum_{i=1}^{n_v} |v_i| \quad (233)$$

$p = 2$ , para:

$$\|v\|_2 = \sqrt{\sum_{i=1}^{n_v} |v_i|^2} \quad (234)$$

$p = \text{valor real positivo}$ , para:

$$\|v\|_p = \left(\sum_{i=1}^{n_v} |v_i|^p\right)^{\frac{1}{p}} \quad (235)$$

$p = \infty$ , para:



$$\|v\|_{\infty} = \max_{1 \leq i \leq n_v} |v_i| \quad (236)$$

$p = -\infty$ , para:

$$\|v\|_{-\infty} = \min_{1 \leq i \leq n_v} |v_i| \quad (237)$$

#### 4.15.14. licols

Esta rotina de código sem loop encontra um subconjunto máximo de colunas linearmente independentes em uma matriz (MATT, 2020).

---

**Função** licols

**Início**

**Entrada**

X //Matriz de dados das variáveis regressoras  
tol //tolerância estimada. Padrão =  $10^{10}$

**Saída**

Xsub //Colunas extraídas de X  
idx //índices das colunas extraídas

**Determine** //mínimo entre tolerância dada e a tolerância padrão

**Faça** //decomposição QR de X

**Recupere** //diagonal da matriz triangular superior R e a matriz de permutação.  $AE = QR$ , onde o fator R é uma matriz triangular superior  $m \times n$ , Q é uma matriz ortogonal  $m \times n$  e E é a matriz de permutação

**Estime** //classificação de X fazendo  $r \leftarrow$  elementos da diagonal de R que são maiores ou iguais à tolerância

idx  $\leftarrow$  sort(E(1:r))

Xsub  $\leftarrow$  X(:,idx)

---

## 5. RESULTADOS E DISCUSSÃO

Nesta seção serão discutidos os resultados obtidos após a aplicação da metodologia descrita na seção 4. A aplicação foi feita em três exemplos diferentes e em cada um desses exemplos foram testados quatro casos distintos.

Os três exemplos foram: Aplicação em equações matemáticas de difícil resolução; Aplicação em uma coluna de destilação simulada em Aspen Plus; Aplicação em uma planta de tratamento de efluentes simulada no software Simulink.

Para cada uma das aplicações, serão tratados quatro casos diferentes para apresentar a efetividade do algoritmo. Os quatro casos são apresentados a seguir.

- **Caso 1:** 80% do total de amostras permitidas por resposta em processamento são utilizados para gerar metamodelos através da técnica de regressão linear por mínimos quadrados. Os demais 20% são utilizados para geração de metamodelos usando métodos não lineares de regressão gaussiana.
- **Caso 2:** Mesma situação do caso 1, porém utilizando a metodologia Lola-Voronoi para coletar informações da não-linearidade da resposta em processamento.
- **Caso 3:** 100% do total de amostras permitidas por resposta em processamento são utilizadas para gerar metamodelos através da técnica de regressão linear dos mínimos quadrados. Vale ressaltar que respostas no conjunto das não-convergadas são processadas em background exclusivamente por modelos não-lineares.
- **Caso 4:** 100% do total de amostras permitidas por resposta em processamento são utilizadas para geração de metamodelos usando métodos não lineares de regressão gaussiana. É importante notar que, como os regressores são sempre gerados pelo método de regressão dos mínimos quadrados, este método também é considerado como candidato a metamodelo.

Os resultados consistem na apresentação dos gráficos de teste dos metamodelos com dados diferentes dos usados na construção. Os dados de

teste foram criados pelo método de amostragem Latin Hipercubo, técnica apresentada na seção 2.2.1.9. Neste caso, são comparadas as respostas do metamodelo com os valores das respostas obtidos a partir das simulações do modelo real. Também serão apresentados os gráficos da evolução do processo iterativo na construção de cada metamodelo em relação as variáveis de importância como o desvio do Erro (SER) e o valor do  $Q^2$ .

Outro resultado importante é medição da eficiência do algoritmo feedback inclusivo, em outras palavras, pretende-se também aferir o custo computacional para a geração dos metamodelos. As métricas consideradas para este fim são baseadas em parâmetros de desempenho e são apresentadas a seguir:

- Tempo de processamento do algoritmo para construção de metamodelos. Todos os casos foram processados em um computador Dell Precision 3630 Tower com processador Intel Xeon E-2124G com CPU de 3.40GHz e 16GB de memória RAM.
- Percentual de metamodelos que convergiram por um dos dois critérios definidos no algoritmo. Note que isso não significa que um determinado metamodelo atingiu um  $Q^2$  maior ou igual ao limite mínimo estabelecido.
- Percentual de metamodelos que atingiram um  $Q^2$  maior ou igual ao limite mínimo estabelecido. Note que mesmo atingindo tal marca, isso não significa afirmar que um determinado metamodelo convergiu.
- Número total de amostras utilizadas para construção do metamodelo. Essa métrica é de extrema importância pois afere uma das principais propostas do algoritmo que é a capacidade de construir um metamodelo com o menor número de amostras possível.
- Número mínimo de amostras usado na construção de um determinado metamodelo. Isso representa a capacidade do algoritmo em detectar respostas “fáceis” de convergir.

### **5.1. Aplicação da metodologia em equações matemáticas diretamente no MATLAB**

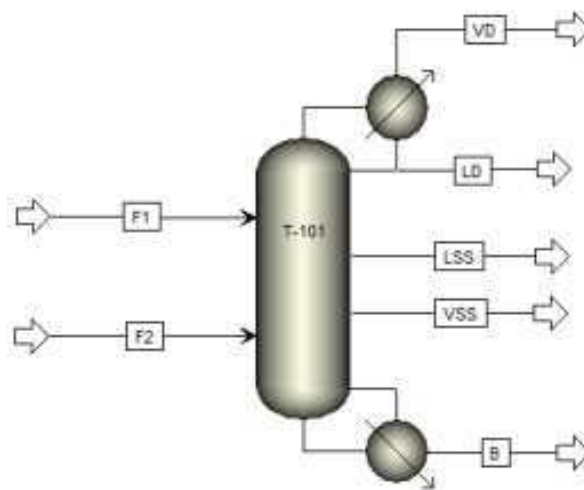
Para testar a metodologia proposta, o algoritmo foi aplicado a um conjunto de oito equações matemáticas a fim de encontrar um modelo substituto para cada uma delas.

As equações matemáticas utilizadas são de difícil resolução e foram apresentadas na seção 4.15.9 deste trabalho. A numeração das equações no algoritmo segue a ordem a qual as equações foram apresentadas em sua respectiva seção. A ordem é:

- 1 – Rastrigin;
- 2 – Schwefel;
- 3 – Hiperelipsoide rotacionada;
- 4 – Styblinski-Tang;
- 5 – Zhakarov;
- 6 – Soma dos quadrados;
- 7 – Ackley;
- 8 – Autorial.

### **5.2. Aplicação da metodologia em uma planta de destilação em Aspen Plus**

A Figura 10 apresenta o esquema da coluna de destilação no software Aspen Plus. Essa coluna de destilação foi proposta por Seader, Henley e Roper (2011).



**Figura 11: Coluna de destilação no Aspen Plus**  
**Fonte: Próprio Autor. Simulação no Aspen**

Os componentes na entrada da coluna de destilação são: Etano, Propano, N-butano, N-pentano e N-hexano. As Tabelas 2 e 3 apresentam as especificações das correntes F1 e F2:

**Tabela 1: Especificações da corrente F1**  
**Fonte: Próprio Autor**

| Variável                | Valor e unidade |
|-------------------------|-----------------|
| Temperatura             | 170 °F          |
| Pressão                 | 300 psia        |
| Vazão molar (Etano)     | 2,5 lbmol/h     |
| Vazão molar (Propano)   | 14 lbmol/h      |
| Vazão molar (N-butano)  | 19 lbmol/h      |
| Vazão molar (N-pentano) | 5 lbmol/h       |
| Vazão molar (N-hexano)  | 0,5 lbmol/h     |

**Tabela 2: Especificações da corrente F2**  
**Fonte: Próprio autor**

| Variável                | Valor e unidade |
|-------------------------|-----------------|
| Temperatura             | 230 °F          |
| Pressão                 | 275 psia        |
| Vazão molar (Etano)     | 0,5 lbmol/h     |
| Vazão molar (Propano)   | 6 lbmol/h       |
| Vazão molar (N-butano)  | 18 lbmol/h      |
| Vazão molar (N-pentano) | 30 lbmol/h      |
| Vazão molar (N-hexano)  | 4,5 lbmol/h     |

A coluna possui 16 estágios, condensador do tipo parcial – líquido – vapor. A razão de destilado é 20 lbmol/h e a razão de refluxo é 29. As alimentações F1

e F2 entram na coluna nos estágios 6 e 9, respectivamente. Existe uma retirada de líquido no estágio 3 igual a 3 lbmol/h e uma retirada de vapor no estágio 13 igual a 37 lbmol/h.

O perfil de pressão na coluna é de 238 psia no condensador, 240 psia no estágio 2 e uma queda de pressão de 0,2 psia em cada estágio. A fração de vapor de destilado no condensador é igual a 0,75 molar.

Para a utilização do algoritmo na determinação do metamodelo que descreve a coluna de destilação, é necessário especificar as variáveis de entrada e saída do processo.

As variáveis de entrada são: razão destilado/alimentação (D:F); Razão de refluxo (RR); Fração de destilado/vapor (RDV); razão molar da corrente de vapor lateral para corrente de alimentação (VSS:F); Razão molar da corrente líquida lateral para corrente de alimentação (LSS:F).

As variáveis de saída ou variáveis resposta são:

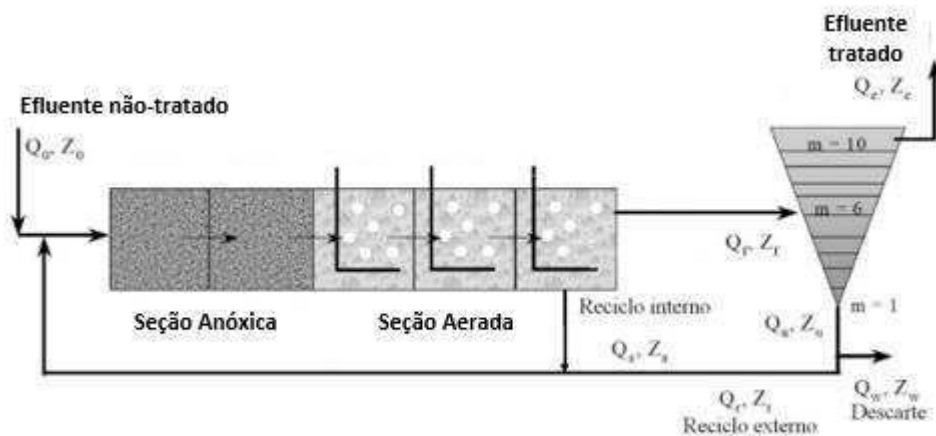
- 1 - Vazão molar de n-butano na corrente de fundo;
- 2 - Vazão molar de n-pentano na corrente de fundo;
- 3 - Vazão molar de etano na corrente de destilado líquido;
- 4 - Vazão molar de propano na corrente de destilado líquido;
- 5 - Vazão molar do etano na corrente de destilado vapor;
- 6 - Vazão molar de propano na corrente de destilado vapor;
- 7 - Vazão molar de propano na corrente de retirada lateral líquida;
- 8 - vazão molar de butano na corrente de retirada lateral líquida;
- 9 - Vazão molar de butano na corrente de saída lateral vapor.

### **5.3. Aplicação da metodologia a uma planta de tratamento de efluentes em Simulink**

O algoritmo foi aplicado a uma unidade reacional de tratamento de efluentes modelada por Jeppson et al., (2011), denominada BSM1, com o

objetivo de encontrar um metamodelo representativo da mesma. O modelo da planta foi construído utilizando o simulador Simulink.

Basicamente, a planta de tratamento de efluentes consiste em um reator de lodo ativado composto por cinco compartimentos. A Figura 11 apresenta o esquema do processo de tratamento.



**Figura 12: Esquema da planta de tratamento de efluentes**  
Fonte: Jeppson et al. (2011)

A planta considera uma vazão de efluente em média de 18446 m<sup>3</sup>/dia. Observando a figura do processo, as seções anóxica e aerada (quadrados) representam a seção de reação e o triângulo representa a seção de sedimentação. A vazão de descarte corresponde a 385 m<sup>3</sup>/dia. A planta apresenta ainda dois sistemas de reciclos, um com parte do descarte para a alimentação do sistema e outro com parte da saída da seção de reação para a alimentação.

A modelagem possui algumas restrições aqui apresentadas: Temperatura constante; pH constante próximo à neutralidade; Características de entrada do do efluente fixas; Adsorção do substrato instantânea.

As reações que ocorrem no biorreator são: Crescimento aeróbico de heterótrofos; Crescimento anóxico de heterótrofos; Crescimento aeróbico de autótrofos; Decaimento de heterótrofos; Decaimento de autótrofos; Amonificação de nitrogênio solúvel; Hidrólise de orgânicos encapsulados; Hidrólise de nitrogênio encapsulado.

Para avaliar o desempenho da planta, deve-se observar os principais componentes presentes no efluente, como: quantidade de nitrogênio total, demanda química de oxigênio, concentração de amônia, sólidos suspensos e demanda biológica de oxigênio.

Para a utilização do algoritmo na determinação do metamodelo que descreve a planta de tratamento de efluentes, é necessário especificar as variáveis de entrada e saída do processo.

As variáveis de entrada são: reciclo externo ( $Q_r$ ), reciclo interno ( $Q_a$ ), descarte de lodo ( $Q_w$ ), coeficiente de transferência de massa nos compartimentos 3, 4 e 5 ( $k_{la3}$ ,  $k_{la4}$  e  $k_{la5}$ ), equivalente à vazão de ar para estes compartimentos.

As variáveis de saída ou variáveis respostas são:

- 1 - Idade do lodo (sludge age);
- 2 - Razão de microorganismos na alimentação (FM ratio);
- 3 - Demanda química de oxigênio no efluente (COD\_eff);
- 4 - Sólidos suspensos totais no efluente (TSS\_eff);
- 5 - Nitrogênio total no efluente (TN\_eff);
- 6 - Nitrogênio amoniacal no efluente (SNH\_eff);
- 7 - Nitrato e nitrito no efluente (SNO\_eff);
- 8 - Demanda bioquímica de oxigênio no efluente (BOD\_eff).

#### **5.4. Resultados**

Os gráficos mostrados nas Figuras 12 a 23, contêm, para cada resposta, informações acerca do número de amostras usadas para construção do respectivo metamodelo, o  $Q^2$  de validação cruzada resultante do processo de construção, o  $Q^2$  do teste comparativo dos valores da resposta do metamodelo com dados simulados e o tipo de metamodelo final determinado pelo algoritmo.



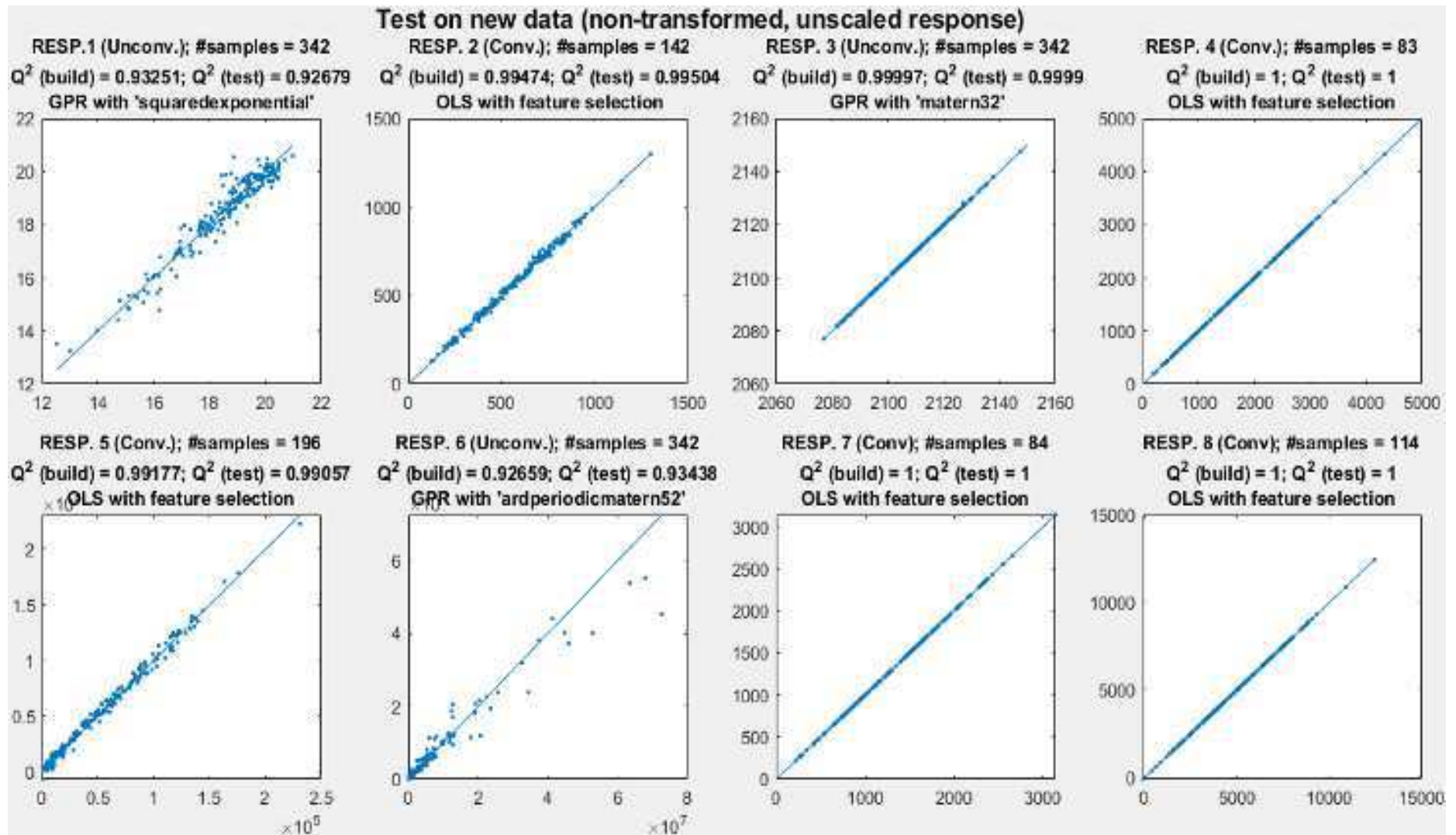


Figura 13: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no MATLAB (Caso 1). Fonte: Próprio Autor.

A Figura 12 apresenta os resultados obtidos para o algoritmo aplicado às equações em Matlab na condição do caso 1. Observa-se que as respostas 2, 4, 5, 7 e 8 convergiram com o  $Q^2$  acima do limite inferior igual a 0,97 e todas pelo método de regressão linear dos mínimos quadrados. As respostas 1, 3 e 6 não convergiram. Apesar da resposta 3 apresentar o  $Q^2$  acima de 0,97, esta resposta não convergiu porque não atingiu o estado estacionário e também não apresentou o valor do  $Q^2$  superior a 0,97 por 20 iterações seguidas.

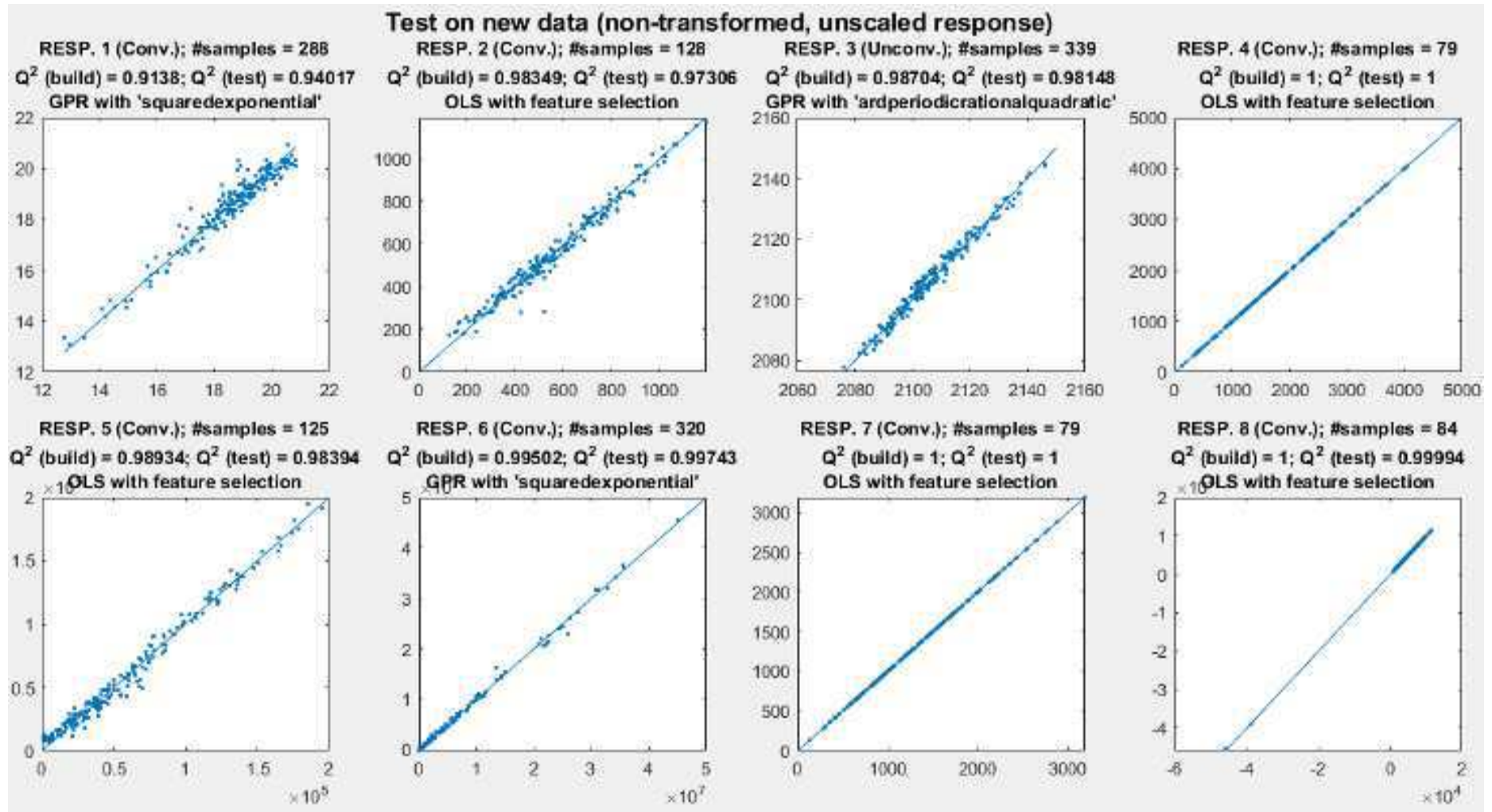


Figura 14: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no MATLAB (Caso 2). Fonte: Próprio Autor.

A Figura 13 apresenta os resultados obtidos para o algoritmo aplicado às equações em Matlab na condição do caso 2, com Lola-Voronoi. Observa-se que as respostas 2, 4, 5, 7 e 8 convergiram pelo método de regressão linear dos mínimos quadrados com  $Q^2$  acima do limite inferior. A resposta 1 convergiu pelo método não linear “squarexponential”. A resposta 1 atingiu o estágio estacionário mas o algoritmo não conseguiu obter um  $Q^2$  acima do limite inferior de 0,97. A resposta 3 não convergiu.

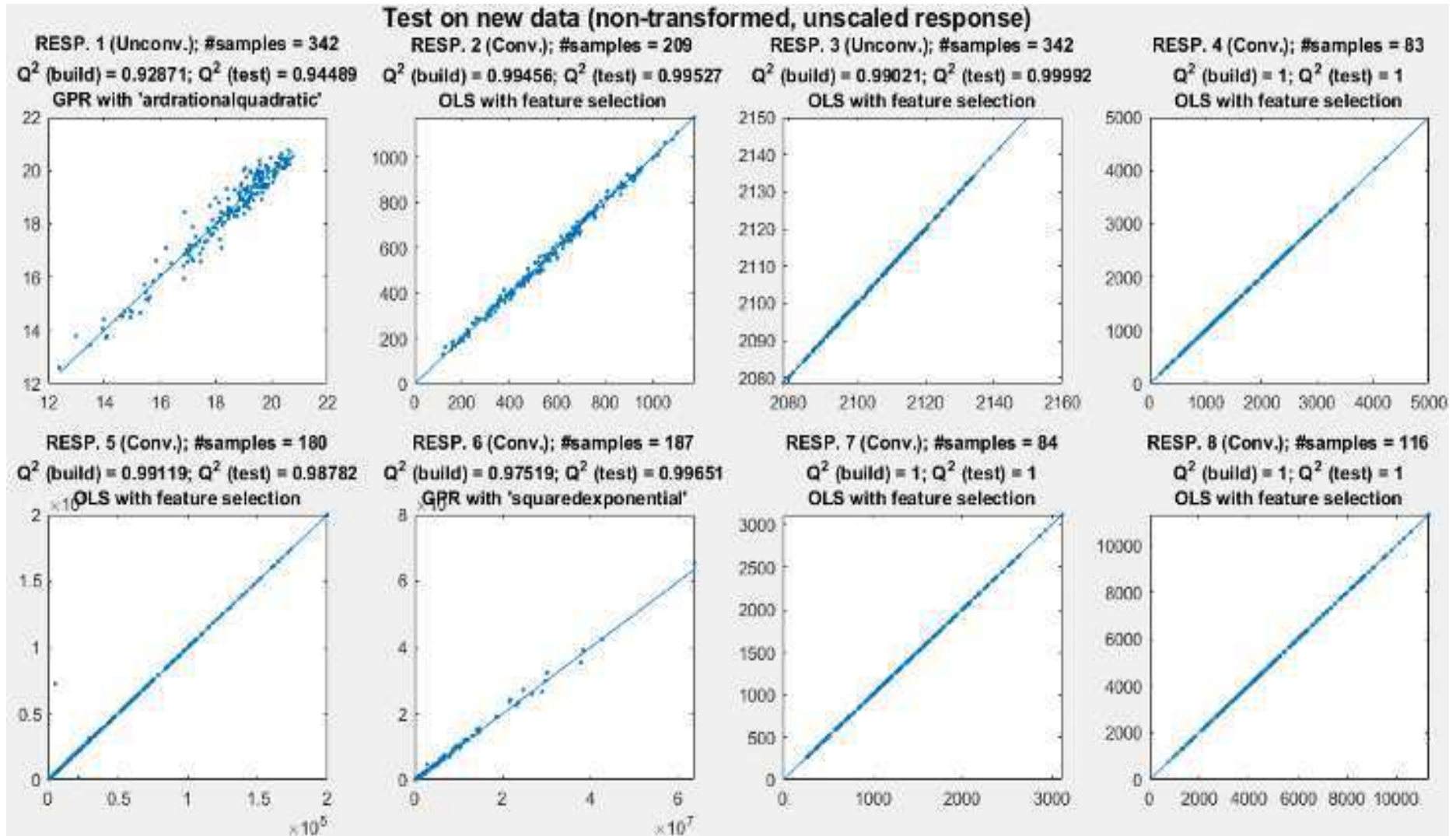


Figura 15: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no MATLAB (Caso 3). Fonte: Próprio Autor.

A Figura 14 apresenta os resultados obtidos para o algoritmo aplicado às equações em Matlab na condição do caso 3, apenas regressão linear. Vale ressaltar que as respostas que são processadas em “background” utilizam apenas o método de regressão não linear. Observa-se que as respostas 2, 4, 5, 7 e 8 convergiram pelo método de regressão linear dos mínimos quadrados e com o valor de  $Q^2$  acima do limite inferior de 0,97. A resposta 6 convergiu pelo método de regressão não linear “squarexponential”. As respostas 1 e 3 não convergiram.

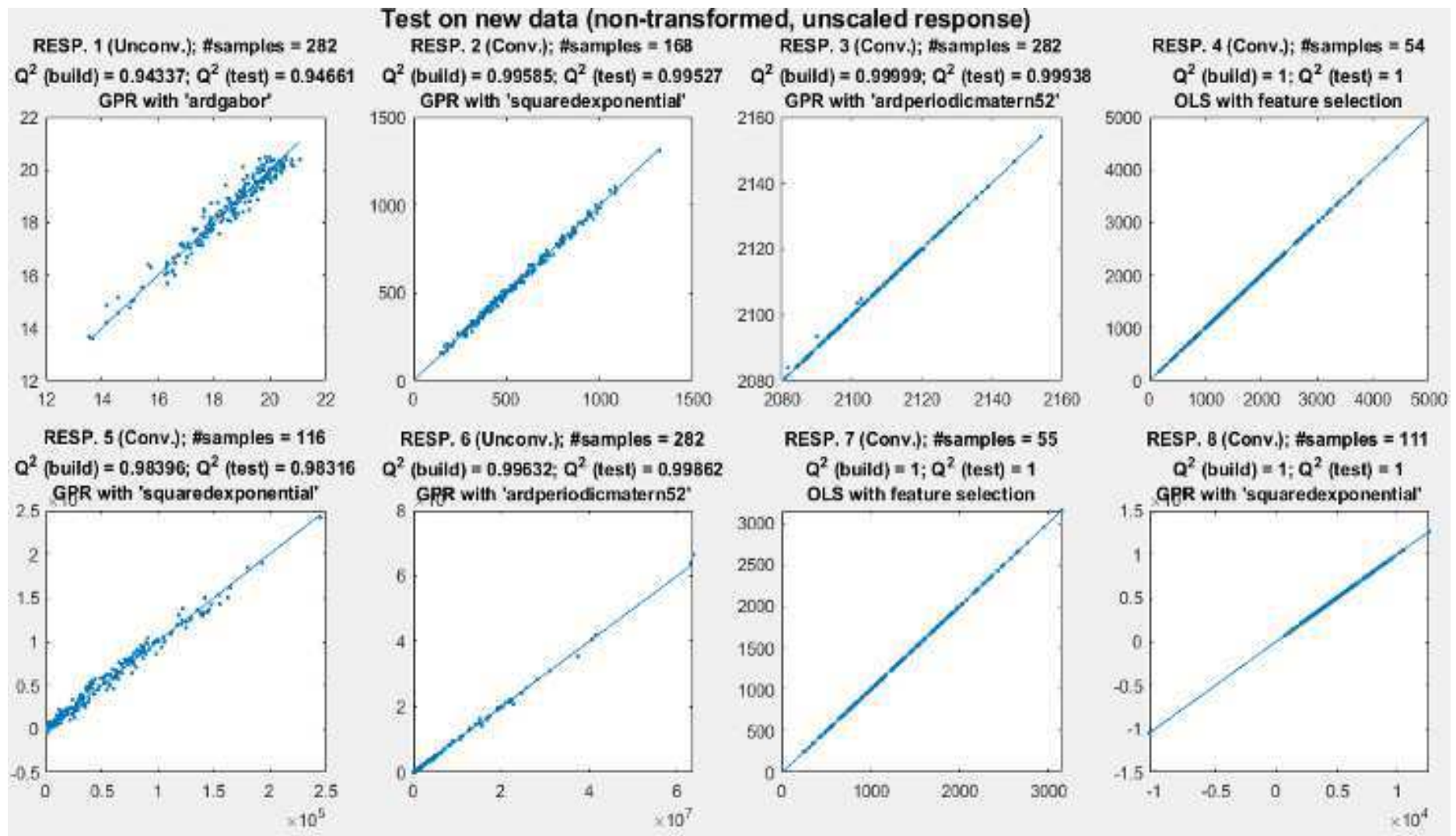


Figura 16: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no MATLAB (Caso 4). Fonte: Próprio Autor.

A Figura 15 apresenta os resultados obtidos para o algoritmo aplicado às equações em Matlab na condição do caso 4, apenas regressão não linear. Como os regressores são selecionados pelo método de regressão não linear, este método também poderá aparecer como resultado para alguma resposta. Observa-se que todas as respostas, exceto a resposta 1, convergiram com o  $Q^2$  acima do limite inferior de 0,97. As respostas 2, 5 e 8 convergiram pelo método de regressão não linear “squarexponential”. As respostas 3 e 6 convergiram pelo método de regressão não linear “ardperiodicmattern52”. As respostas 4 e 7 convergiram pelo método de regressão linear dos mínimos quadrados. A resposta 1 não convergiu.



### Test on new data (non-transformed, unscaled response)

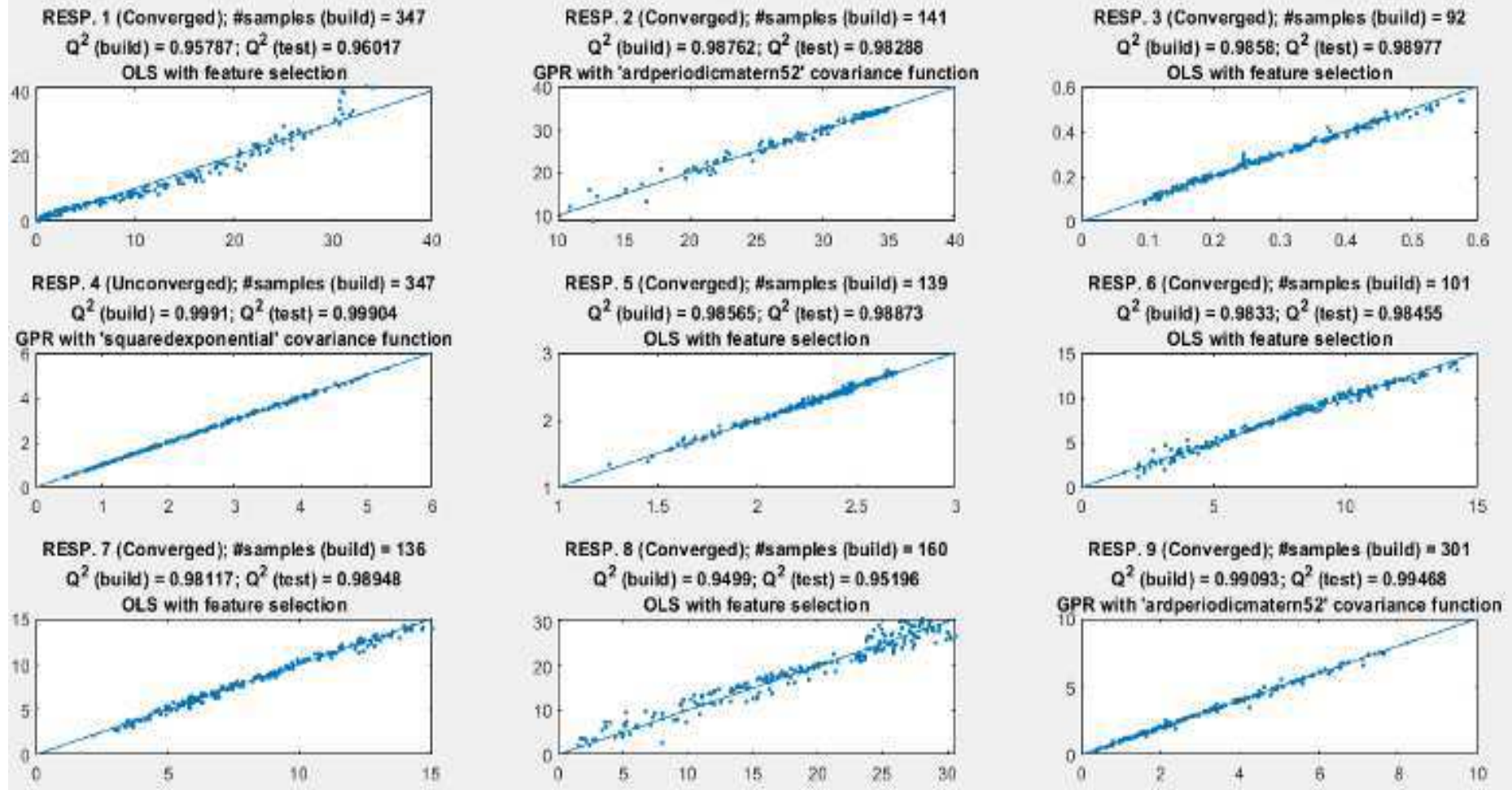


Figura 17: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Aspen Plus (Caso 1). Fonte: Próprio Autor.

A Figura 16 apresenta os resultados obtidos para o algoritmo aplicado à simulação em Aspen Plus na condição do caso 1. Observa-se que as respostas 2, 3, 5, 6, 7 e 9 convergiram com o valor do  $Q^2$  acima do limite inferior de 0,97. As respostas 1 e 8 convergiram com o valor do  $Q^2$  abaixo do limite inferior. As respostas 1, 3, 5, 6, 7 e 8 convergiram pelo método de regressão linear dos mínimos quadrados. As respostas 2 e 9 convergiram pelo método de regressão não linear com a função de covariância “ardperiodicmattern52”. A resposta 4 não atingiu o critério de convergência, embora o valor de  $Q^2$  se encontre acima do limite inferior. Neste caso, cabe ao usuário decidir se usará o metamodelo encontrado ou não.

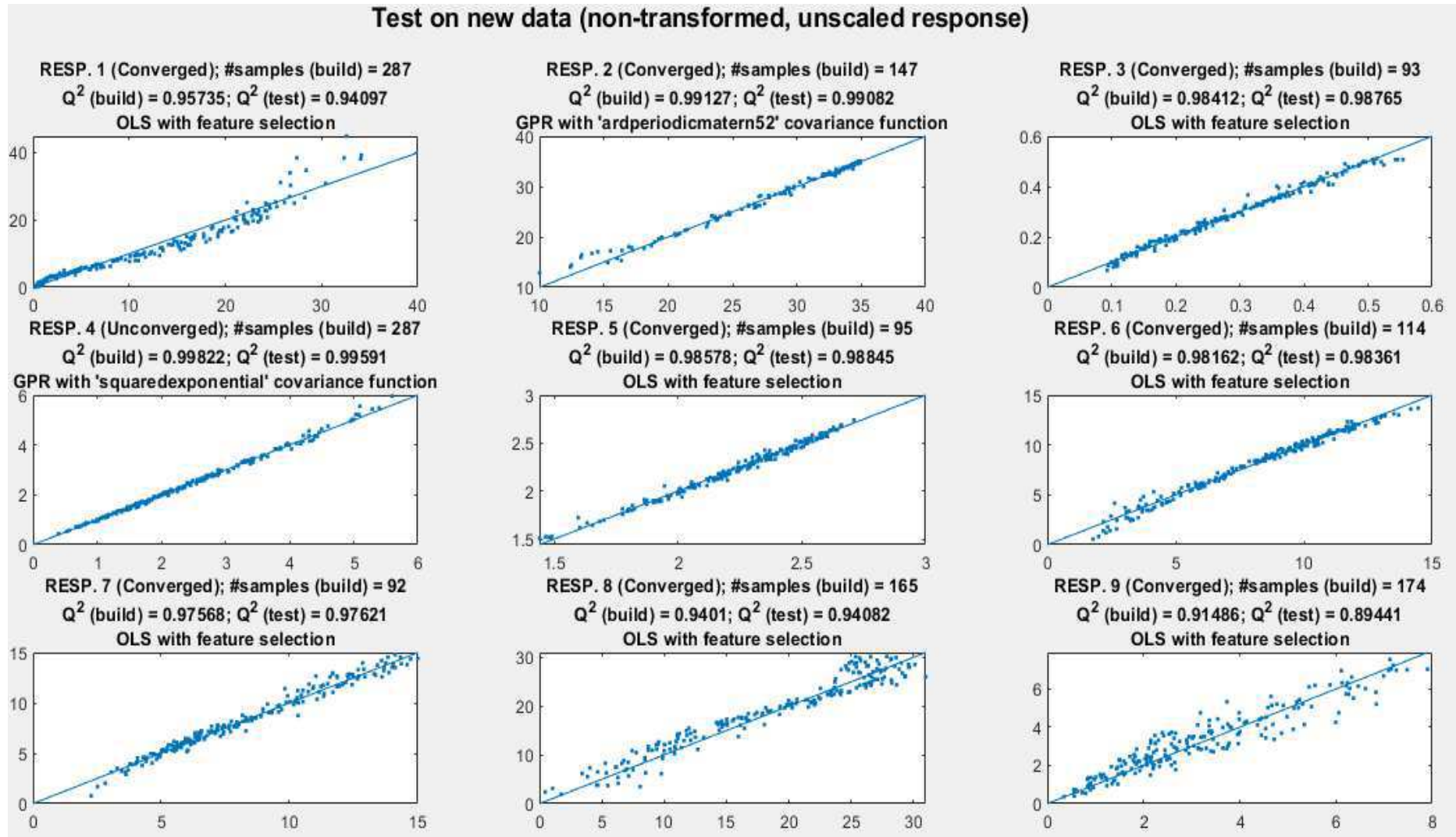


Figura 18: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Aspen Plus (Caso 2). Fonte: Próprio Autor

A Figura 17 apresenta os resultados obtidos para o algoritmo aplicado à simulação em Aspen Plus na condição do caso 2, com Lola-Voronoi. Observa-se que as respostas 2, 3, 5, 6, 7 e 9 convergiram com o valor do  $Q^2$  acima do limite inferior de 0,97. As respostas 1 e 8 convergiram com o valor do  $Q^2$  abaixo do limite inferior. As respostas 1, 3, 5, 6, 7, 8 e 9 convergiram pelo método de regressão linear dos mínimos quadrados. A resposta 2 convergiu pelo método de regressão não linear com a função de covariância “ardperiodicmattern52”. A resposta 4 não atingiu o critério de convergência, embora o valor de  $Q^2$  se encontre acima do limite inferior. Neste caso, cabe ao usuário decidir se usará o metamodelo encontrado ou não.

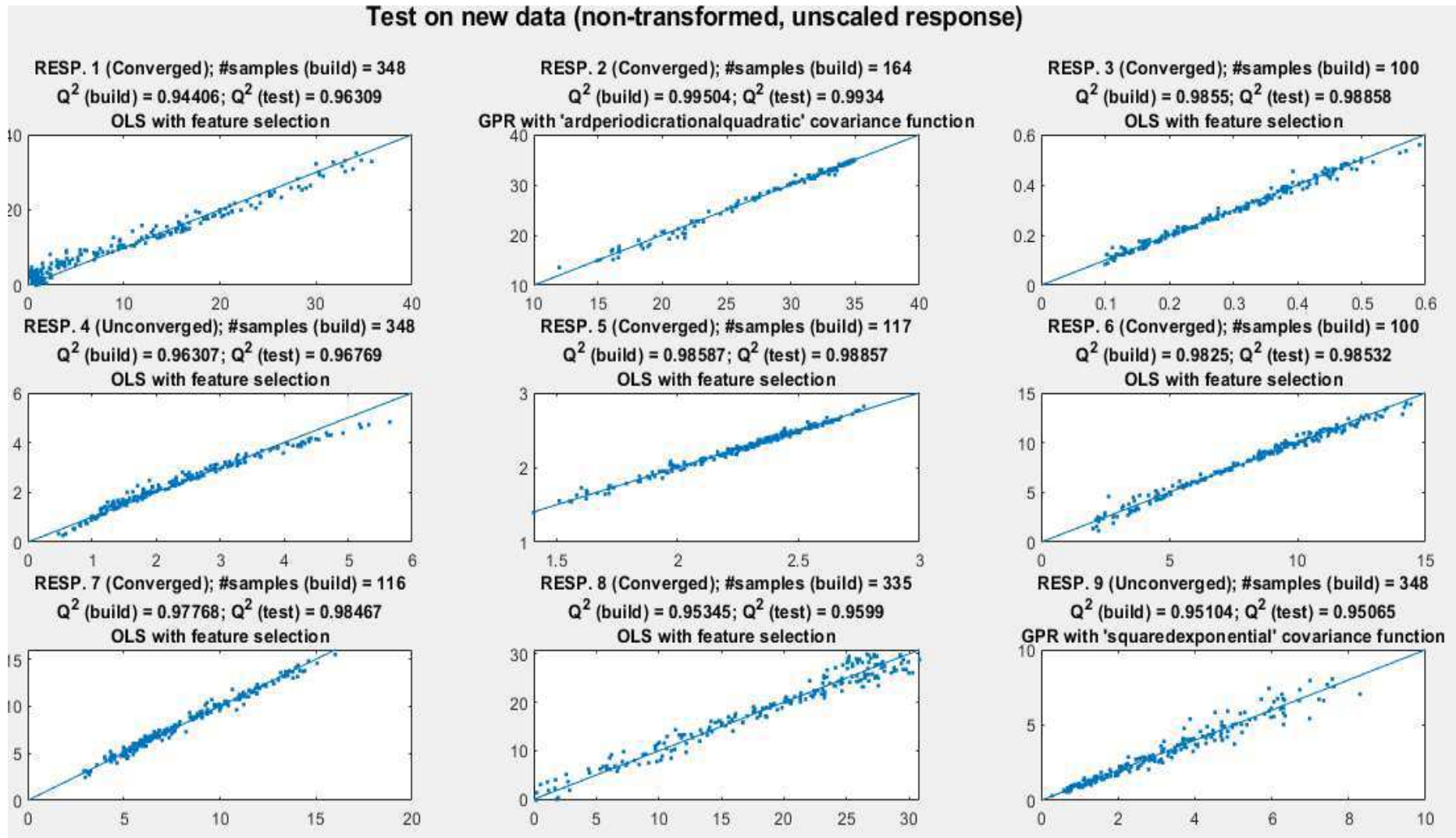


Figura 19: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Aspen Plus (Caso 3). Fonte: Próprio Autor.

A Figura 18 apresenta os resultados obtidos para o algoritmo aplicado à simulação em Aspen Plus na condição do caso 3, apenas regressão linear. Observa-se que as respostas 2, 3, 5, 6, 7 e 9 convergiram com o valor do  $Q^2$  acima do limite inferior de 0,97. As respostas 1 e 8 convergiram com o valor do  $Q^2$  abaixo do limite inferior. As respostas 1, 3, 5, 6, 7 e 8 convergiram pelo método de regressão linear dos mínimos quadrados. As respostas 2 e 9 convergiram pelo método de regressão não linear, sendo a resposta 2 com a função de covariância “ardperiodicmattern52” e a resposta 9 com a função de covariância “squaredexponential”. A resposta 4 não atingiu o critério de convergência.

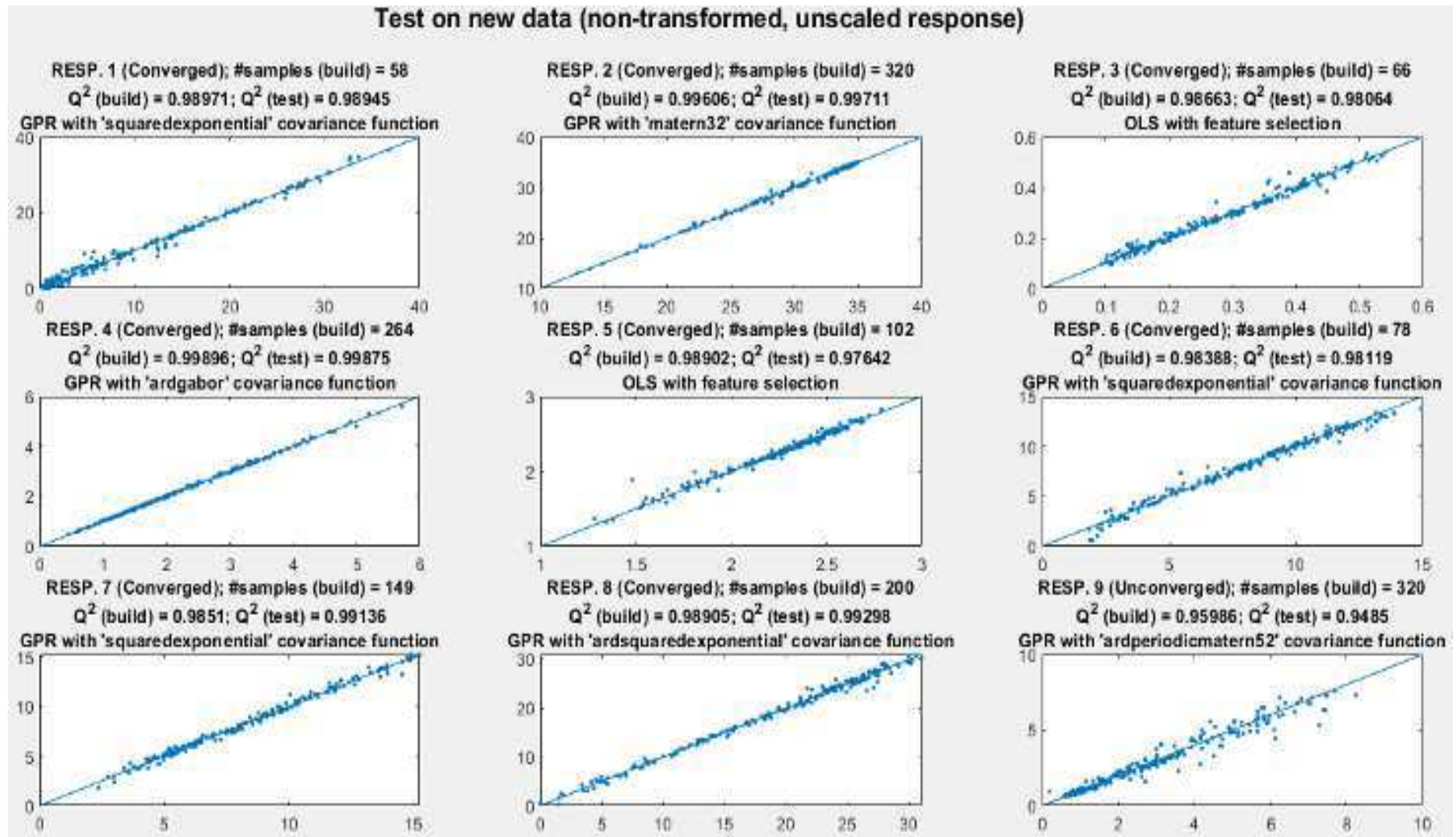


Figura 20: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Aspen Plus (Caso 4). Fonte: Próprio Autor.

A Figura 19 apresenta os resultados obtidos para o algoritmo aplicado à simulação em Aspen Plus na condição do caso 4, apenas regressão não linear. Observa-se que as respostas de 1 a 8 convergiram com o valor do  $Q^2$  acima do limite inferior de 0,97. A resposta 9 convergiu com o valor do  $Q^2$  abaixo do limite inferior. As respostas 3 e 5 convergiram pelo método de regressão linear dos mínimos quadrados. As demais respostas convergiram pelo método de regressão não linear, sendo as respostas 1, 6 e 7 com a função de covariância “squarexponential”, a resposta 2 com a função de covariância “mattern32”, a resposta 4 com a função de covariância “ardgabor”, a resposta 8 com a função de covariância “ardsquarexponential” e a resposta 9 com a função de covariância “ardperiodicmattern52”.



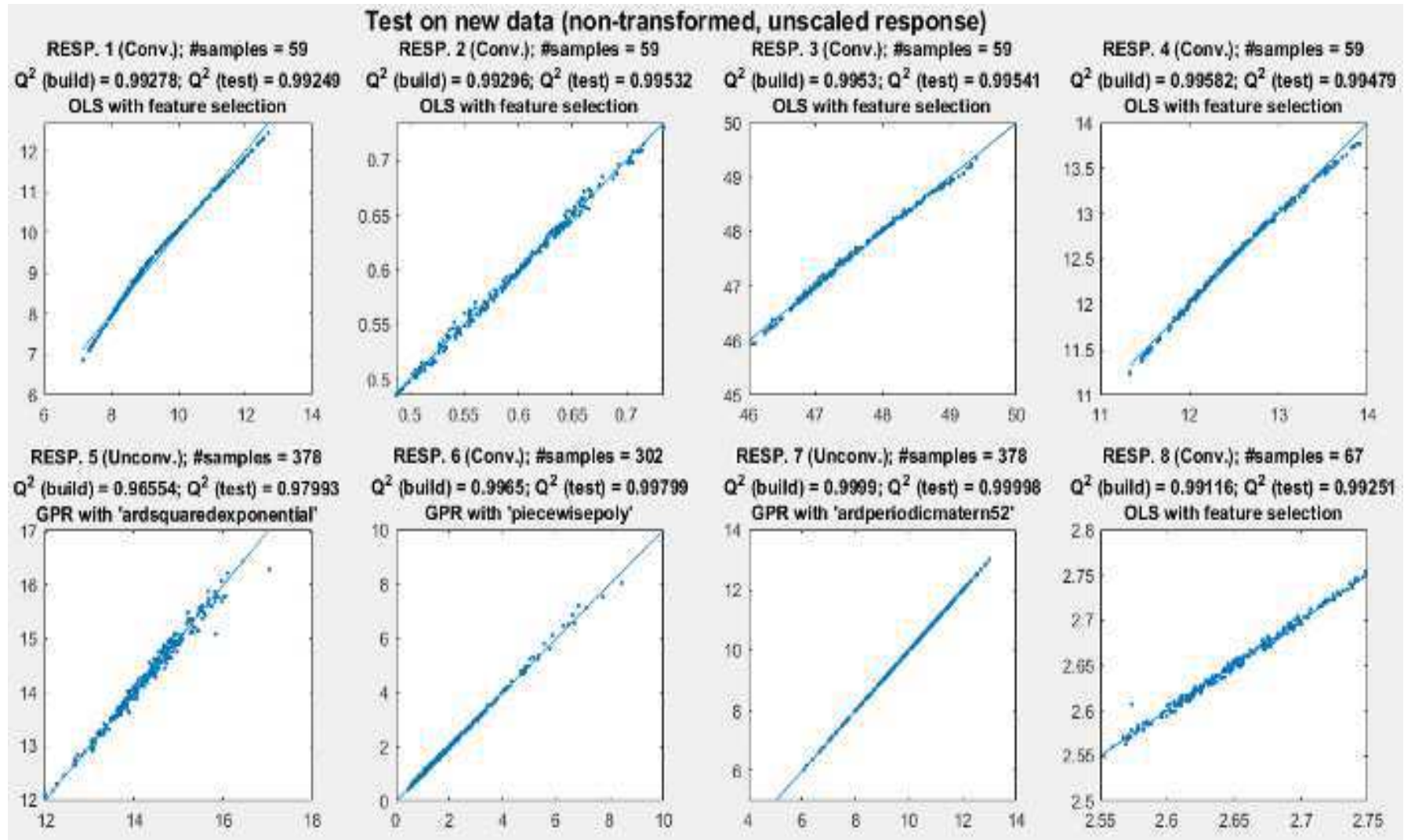


Figura 21: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Simulink (Caso 1). Fonte: Próprio Autor.

A Figura 20 apresenta os resultados obtidos para o algoritmo aplicado à simulação em Simulink na condição do caso 1. Observa-se que todas as respostas, exceto as respostas 5 e 7, convergiram com o valor do  $Q^2$  acima do limite inferior de 0,97. As respostas 1, 2, 3, 4 e 8 convergiram pelo método de regressão linear dos mínimos quadrados. A resposta 6 convergiu pelo método de regressão não linear com função de covariância “placewisepoly”. As respostas 5 e 7 não convergiram, embora a resposta 7 apresente seu valor de  $Q^2$  acima do limite inferior. Neste caso, cabe ao usuário decidir se aceitará o metamodelo ou não.

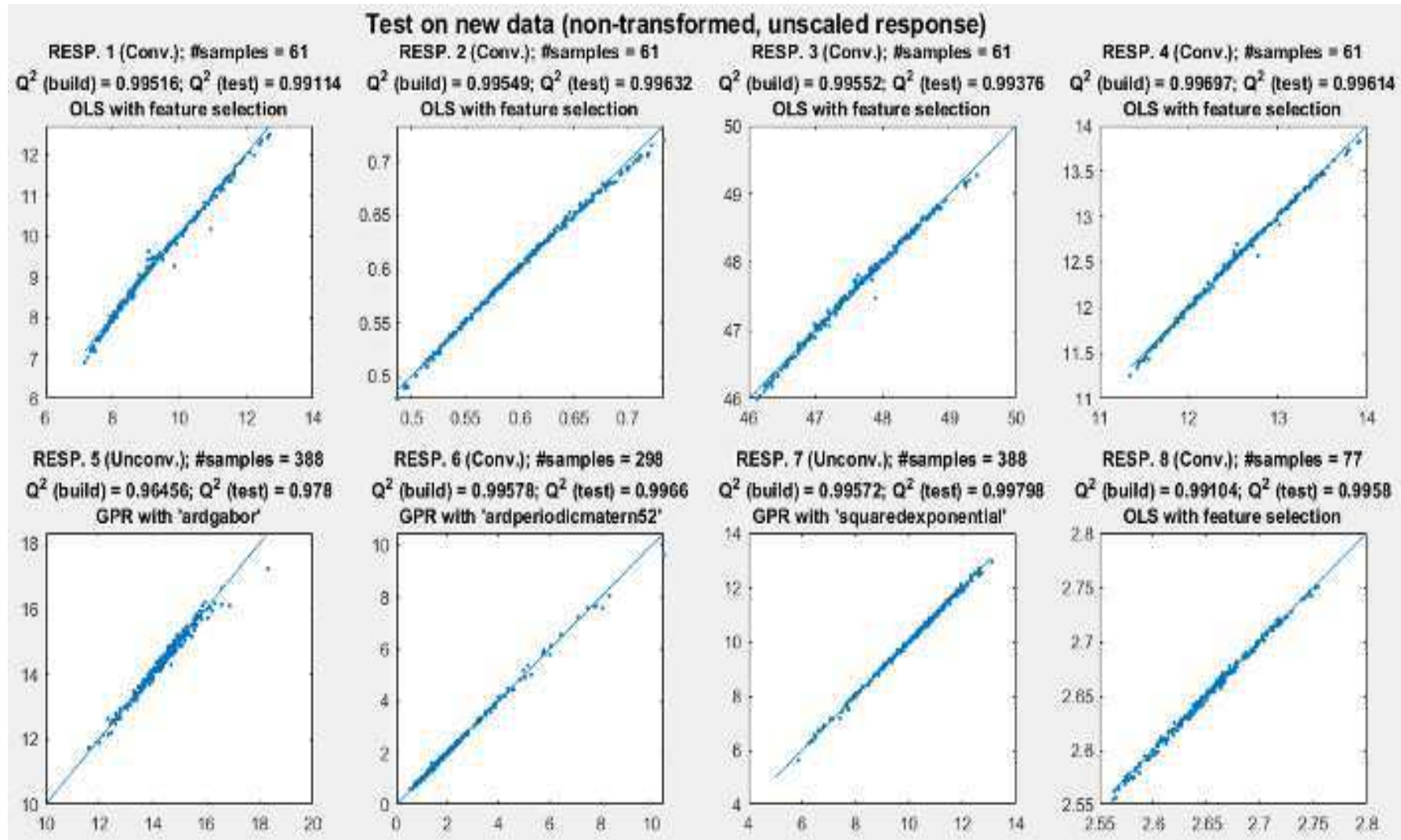


Figura 22: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Simulink (Caso 2). Fonte: Próprio Autor.

A Figura 21 apresenta os resultados obtidos para o algoritmo aplicado à simulação em Simulink na condição do caso 2, com Lola-Voronoi. Observa-se que todas as respostas, exceto as respostas 5 e 7, convergiram com o valor do  $Q^2$  acima do limite inferior de 0,97. As respostas 1, 2, 3, 4 e 8 convergiram pelo método de regressão linear dos mínimos quadrados. A resposta 6 convergiu pelo método de regressão não linear com função de covariância “ardperiodicmattern52”. As respostas 5 e 7 não convergiram, embora a resposta 7 apresente seu valor de  $Q^2$  acima do limite inferior. Neste caso, cabe ao usuário decidir se aceitará os metamodelos ou não.

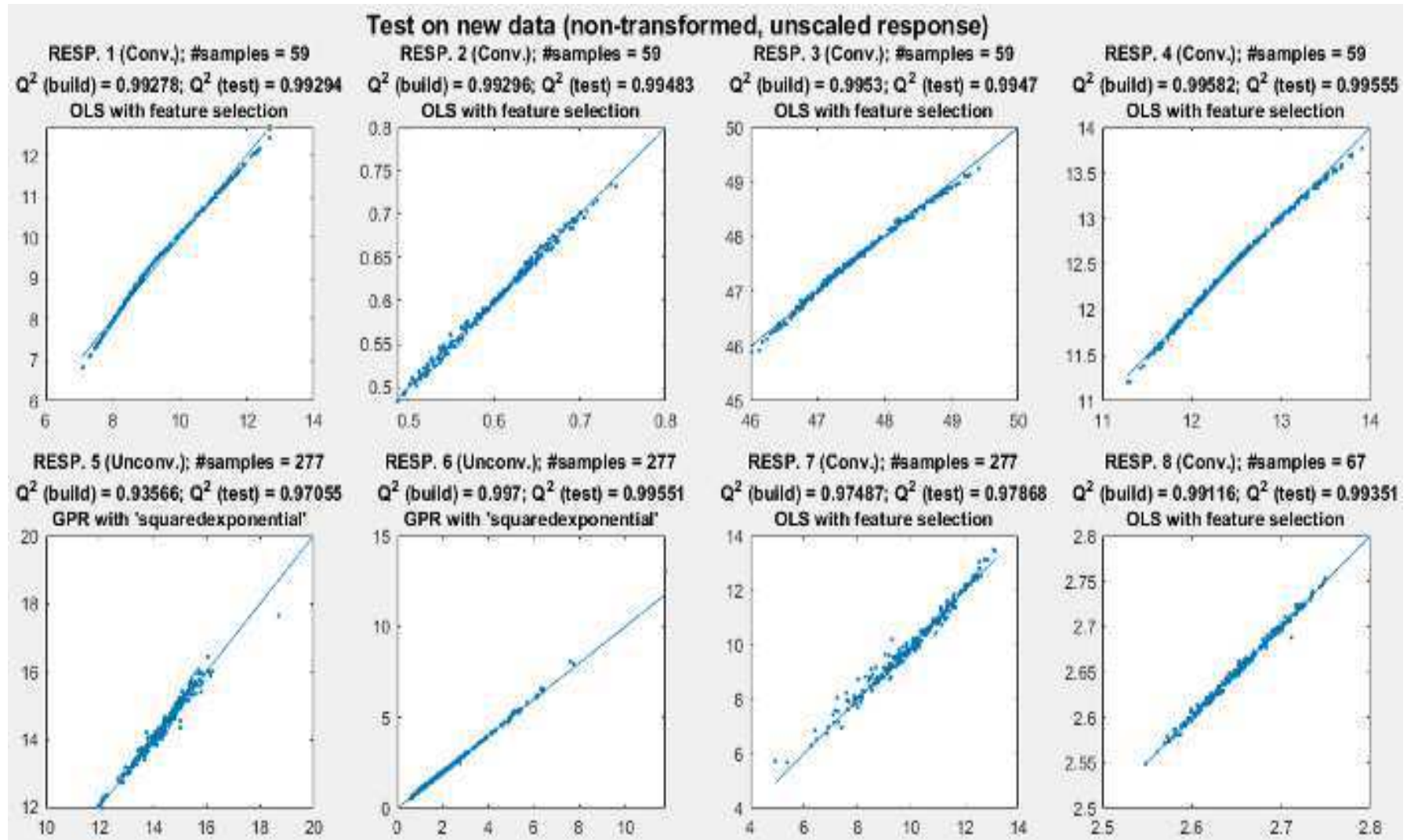


Figura 23: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Simulink (Caso 3). Fonte: Próprio Autor.

A Figura 22 apresenta os resultados obtidos para o algoritmo aplicado à simulação em Simulink na condição do caso 3, apenas regressão linear. Observa-se que todas as respostas, exceto as respostas 5 e 6, convergiram com o valor do  $Q^2$  acima do limite inferior de 0,97. Todas as respostas convergidas foram pelo método de regressão linear dos mínimos quadrados. As respostas 5 e 6 não convergiram mesmo tendo seu valor de  $Q^2$  acima do limite inferior. Neste caso, cabe ao usuário decidir se aceitará os metamodelos ou não.

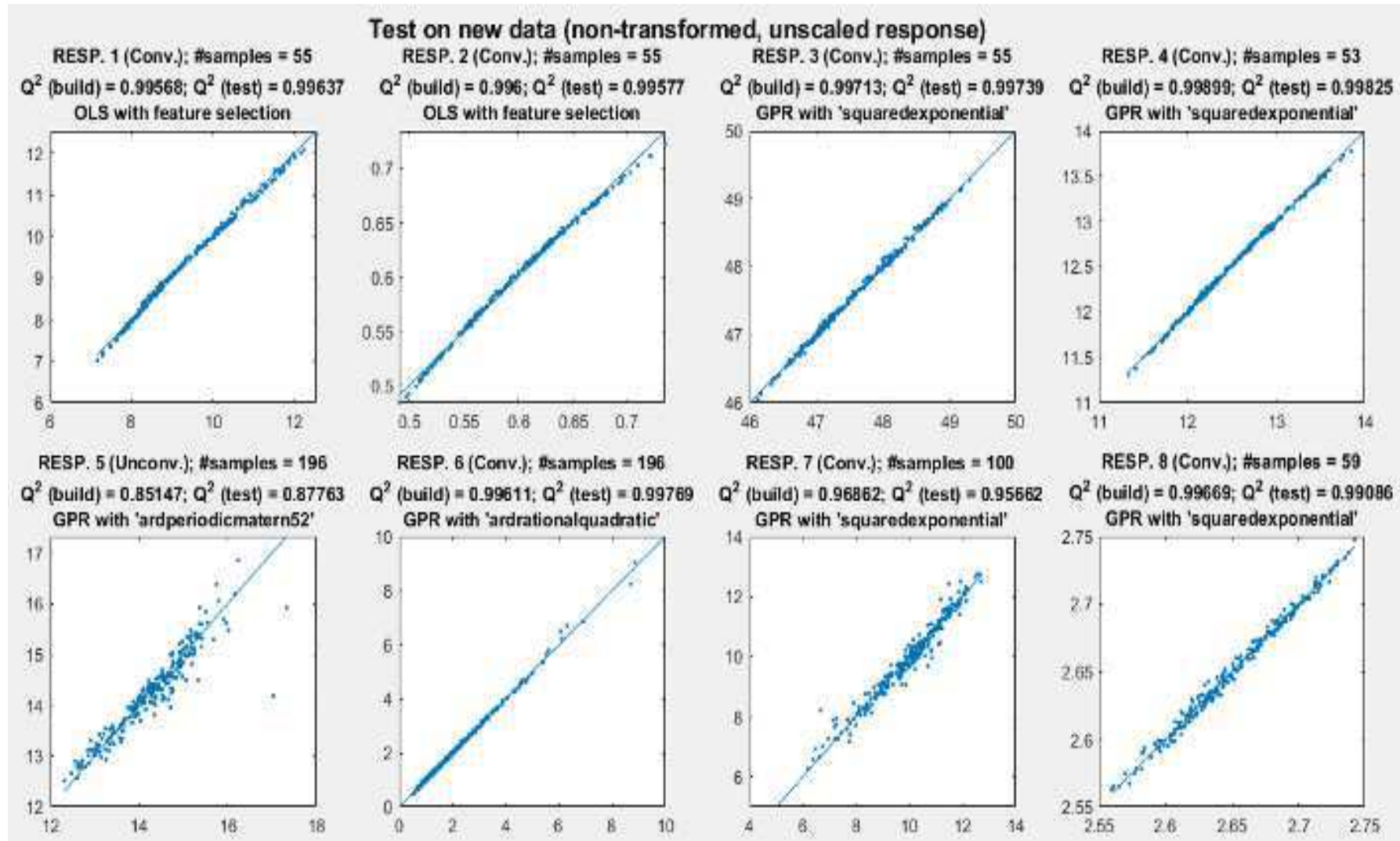


Figura 24: Dados de testes aplicados ao metamodelo comparados com os dados reais de simulação no Simulink (Caso 4). Fonte: Próprio Autor.

A Figura 23 apresenta os resultados obtidos para o algoritmo aplicado à simulação em Simulink na condição do caso 4, apenas regressão não linear. Observa-se que todas as respostas, exceto a respostas 5, convergiram com o valor do  $Q^2$  acima do limite inferior de 0,97. As respostas 1 e 2 convergiram pelo método de regressão linear dos mínimos quadrados. As respostas 3, 4, 6, 7 e 8 convergiram pelo método de regressão não linear, sendo as respostas 3, 4, 7 e 8 com a função de covariância “squarexponential” e a resposta 6 com a função de covariância “ardrationalquadratic”. A resposta 5 não convergiu.

De forma geral percebe-se que a capacidade preditiva de cada metamodelo gerado é bastante significativa. Mesmo metamodelos que não convergiram ou que não ultrapassaram o limite mínimo permitido para o valor de  $Q^2$ , mostram boa capacidade de prever informações não utilizadas para a construção. É possível observar ainda que os valores do  $Q^2$  do processo de construção e do  $Q^2$  do teste comparativo são bem próximos, indicando que a técnica de validação cruzada é realmente eficiente em detectar a capacidade preditiva de um metamodelo. Outro destaque é o aparecimento de metamodelos gerados a partir da regressão linear pelo método dos mínimos quadrados mesmo na modalidade do caso 4. Note que apesar do método de regressão ser linear, os regressores gerados são na grande maioria não-lineares, o que confere não-linearidade ao metamodelo final.

A introdução do método Lola-Voronoi não melhorou significativamente a capacidade preditiva dos metamodelos. Isso provavelmente se deve ao fato de que este método adiciona amostras especificamente para a detecção das não-linearidades da resposta em processamento, o que pode acarretar o adensamento de pontos em regiões específicas que não são de interesse para as outras respostas processadas em “background”.

A evolução do processo iterativo de construção dos metamodelos é mostrada nos gráficos das Figuras 24 a 35, onde são exibidos os valores dos desvios do SER (Erro) e do  $Q^2$  de validação cruzada de cada metamodelo, contando ainda com a adição sequencial de amostras no eixo das abscissas.



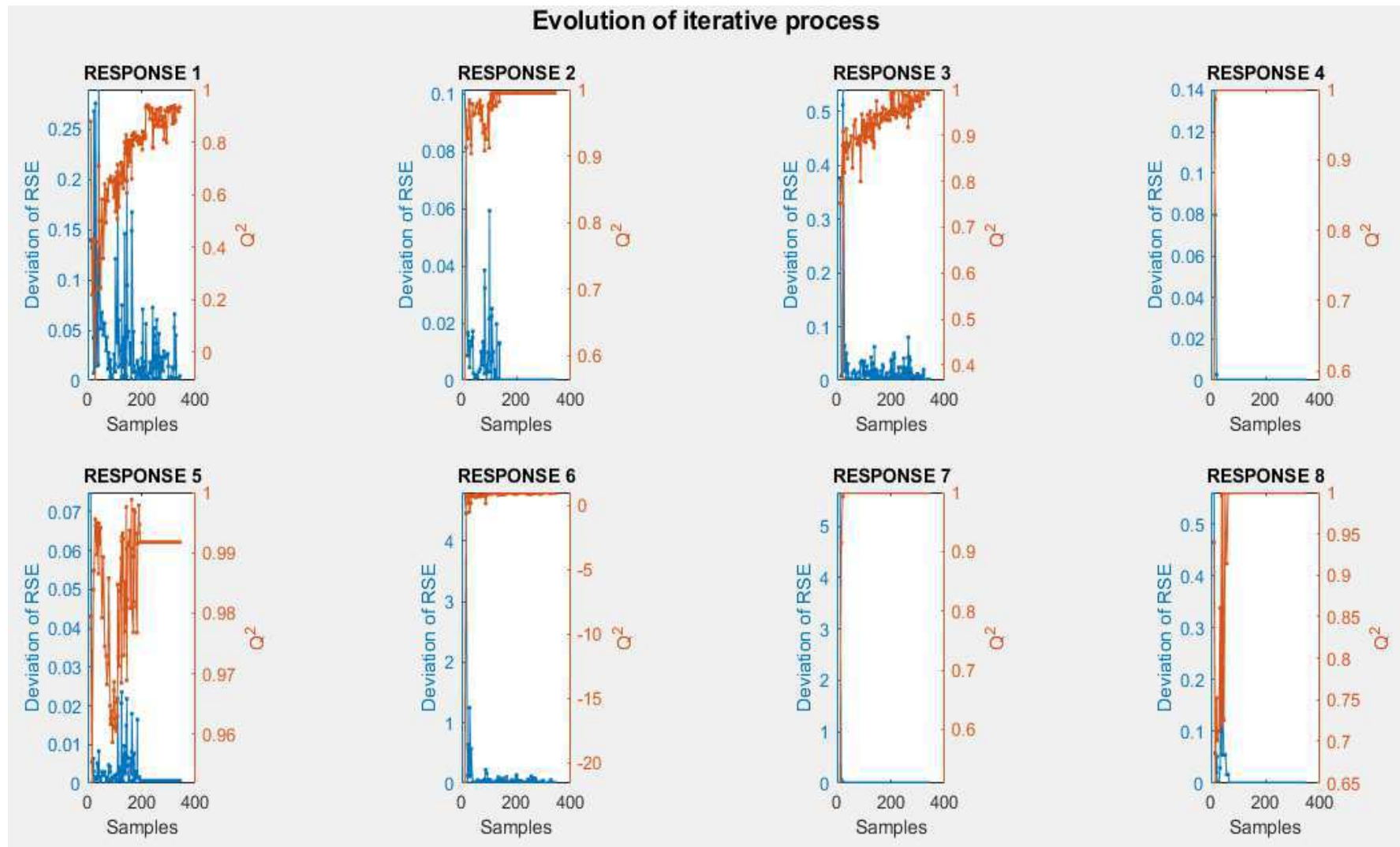


Figura 25: Evolução do processo iterativo do algoritmo aplicado às equações matemáticas em Matlab (Caso 1). Fonte: Próprio Autor

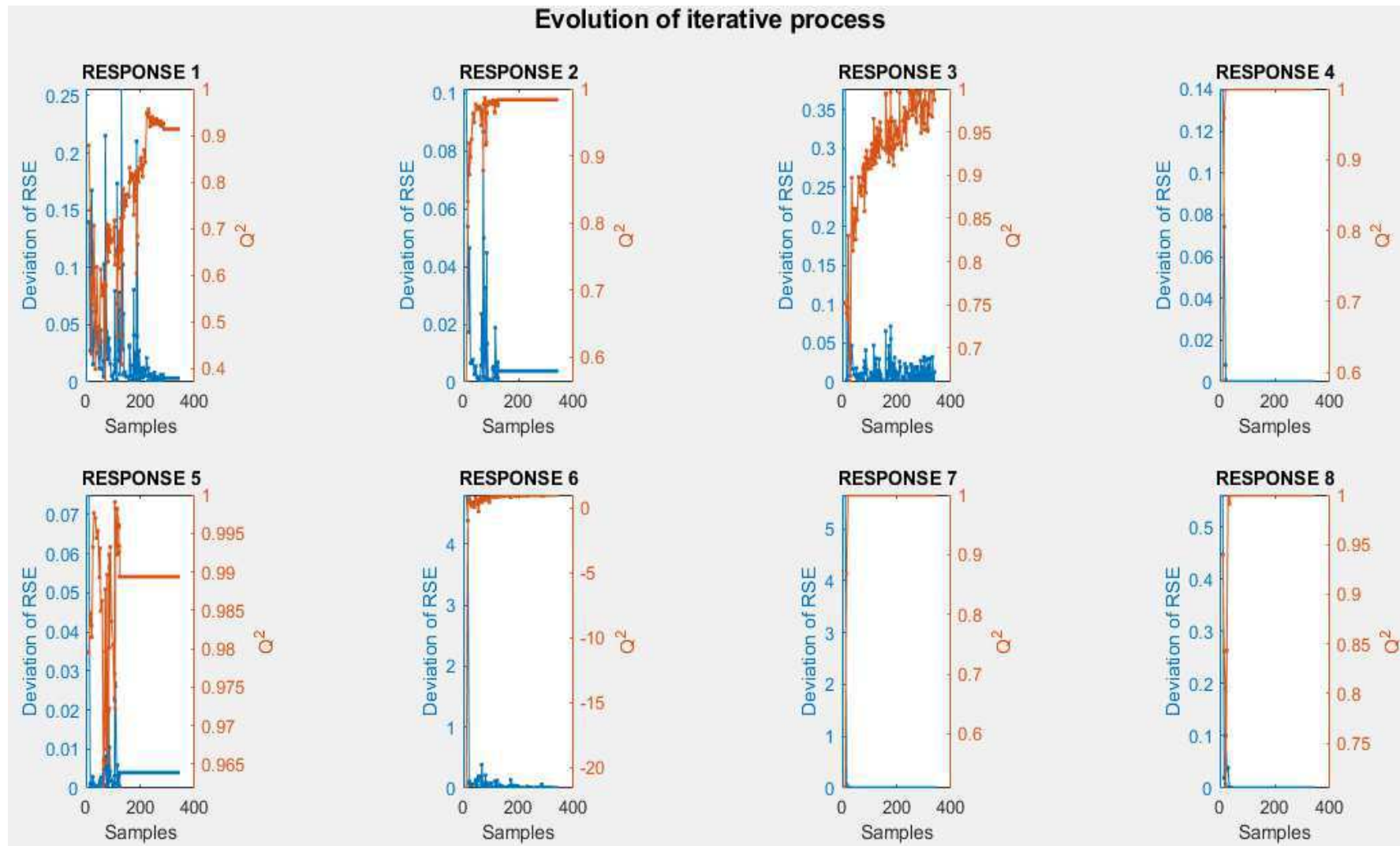


Figura 26: Evolução do processo iterativo do algoritmo aplicado às equações matemáticas em Matlab (Caso 2). Fonte: Próprio Autor

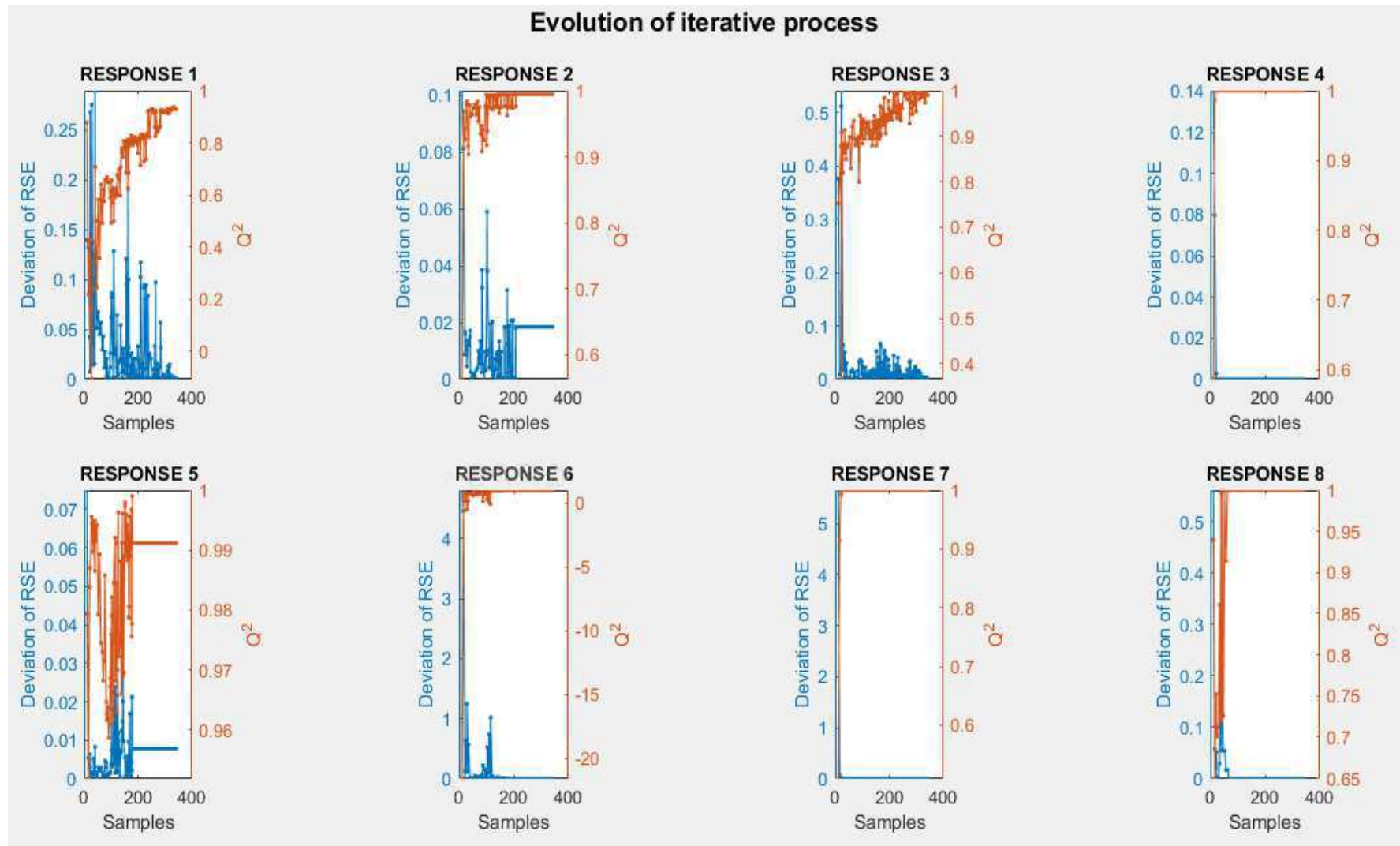


Figura 27: Evolução do processo iterativo do algoritmo aplicado às equações matemáticas em Matlab (Caso 3). Fonte: Próprio Autor

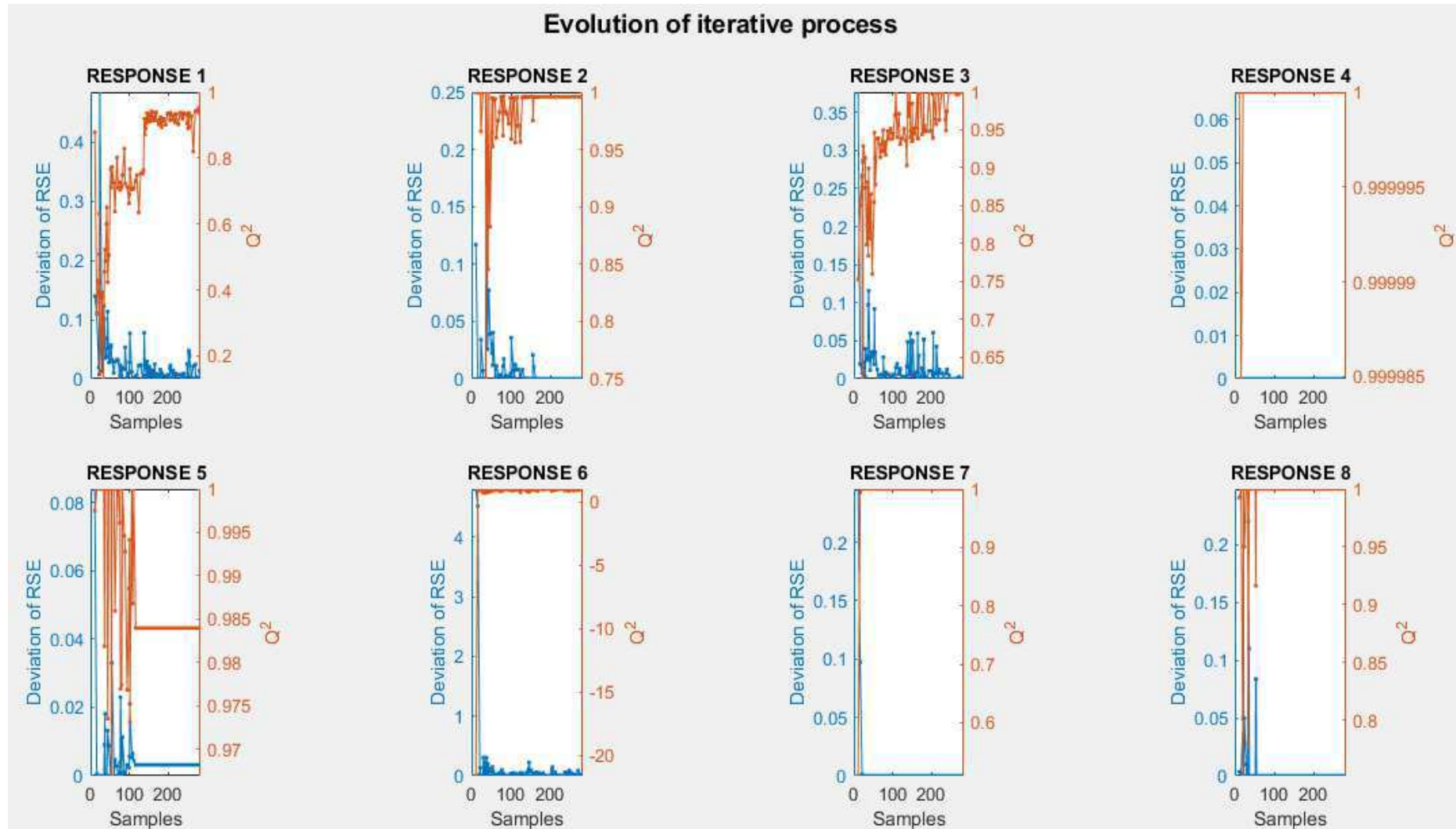


Figura 28: Evolução do processo iterativo do algoritmo aplicado às equações matemáticas em Matlab (Caso 4). Fonte: Próprio Autor

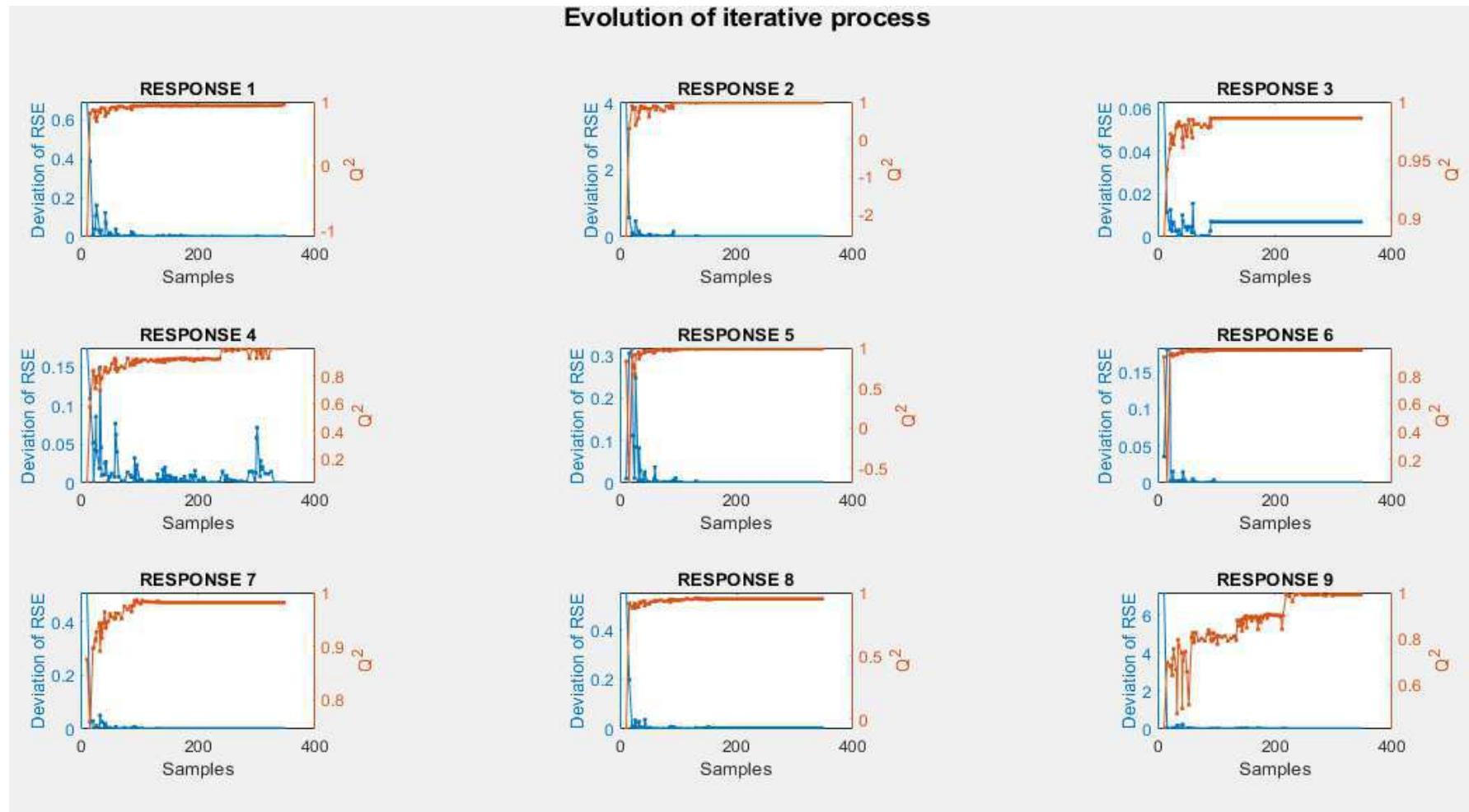
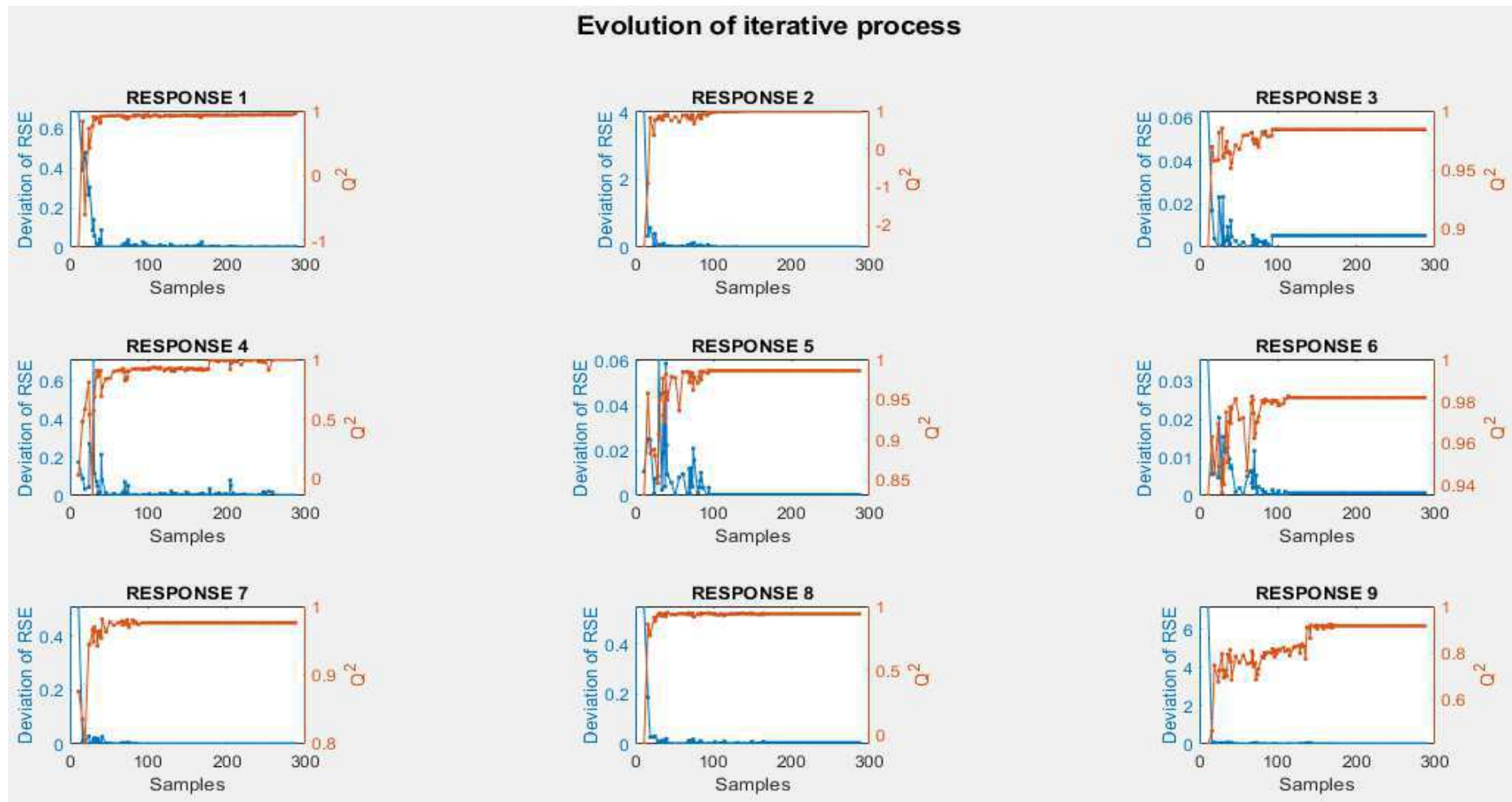


Figura 29: Evolução do processo iterativo do algoritmo aplicado à simulação de uma coluna de destilação em Aspen Plus (Caso 1). Fonte: Próprio Autor



**Figura 30:** Evolução do processo iterativo do algoritmo aplicado à simulação de uma coluna de destilação em Aspen Plus (Caso 2). Fonte: Próprio Autor

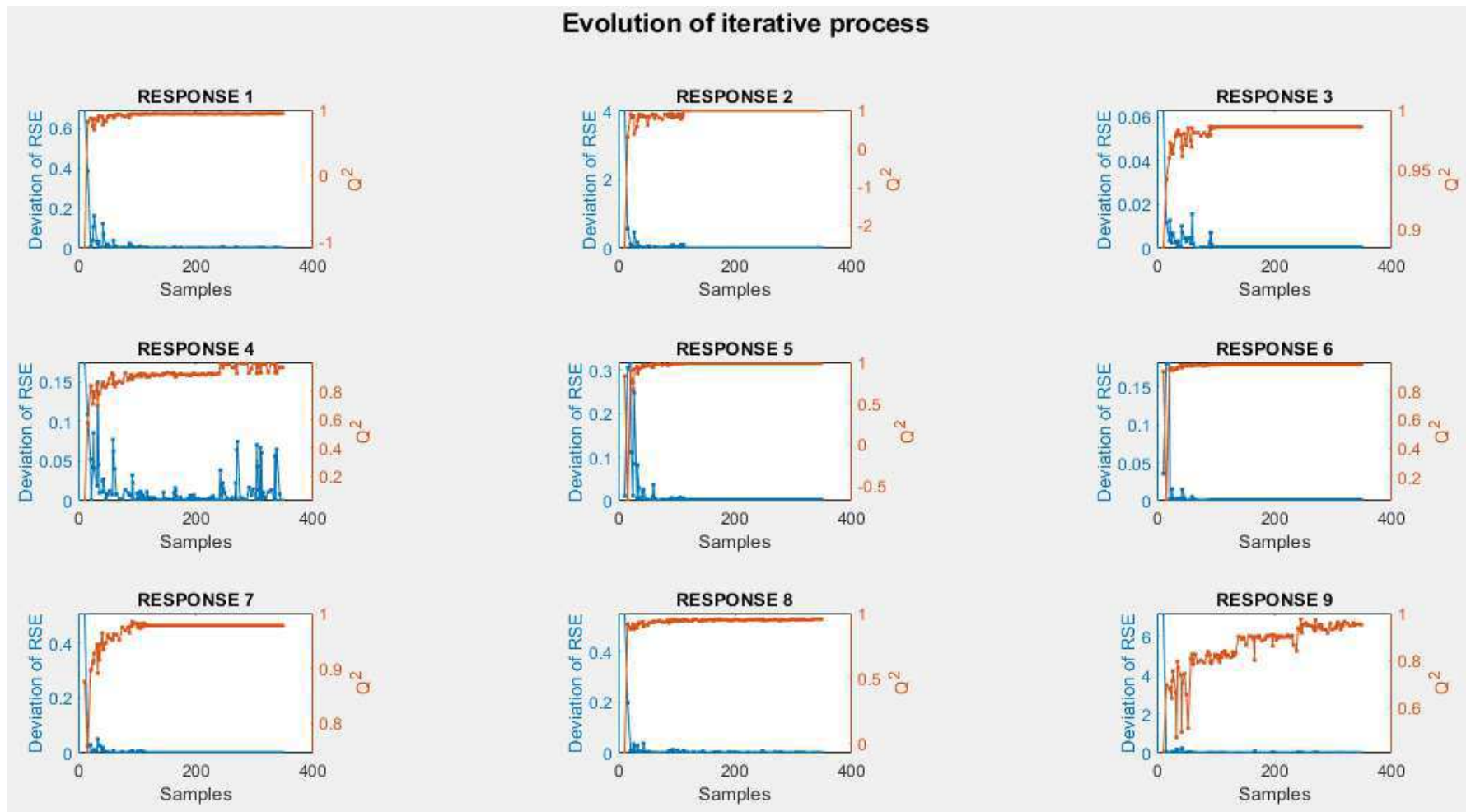


Figura 31: Evolução do processo iterativo do algoritmo aplicado à simulação de uma coluna de destilação em Aspen Plus (Caso 3). Fonte: Próprio Autor

### Evolution of iterative process

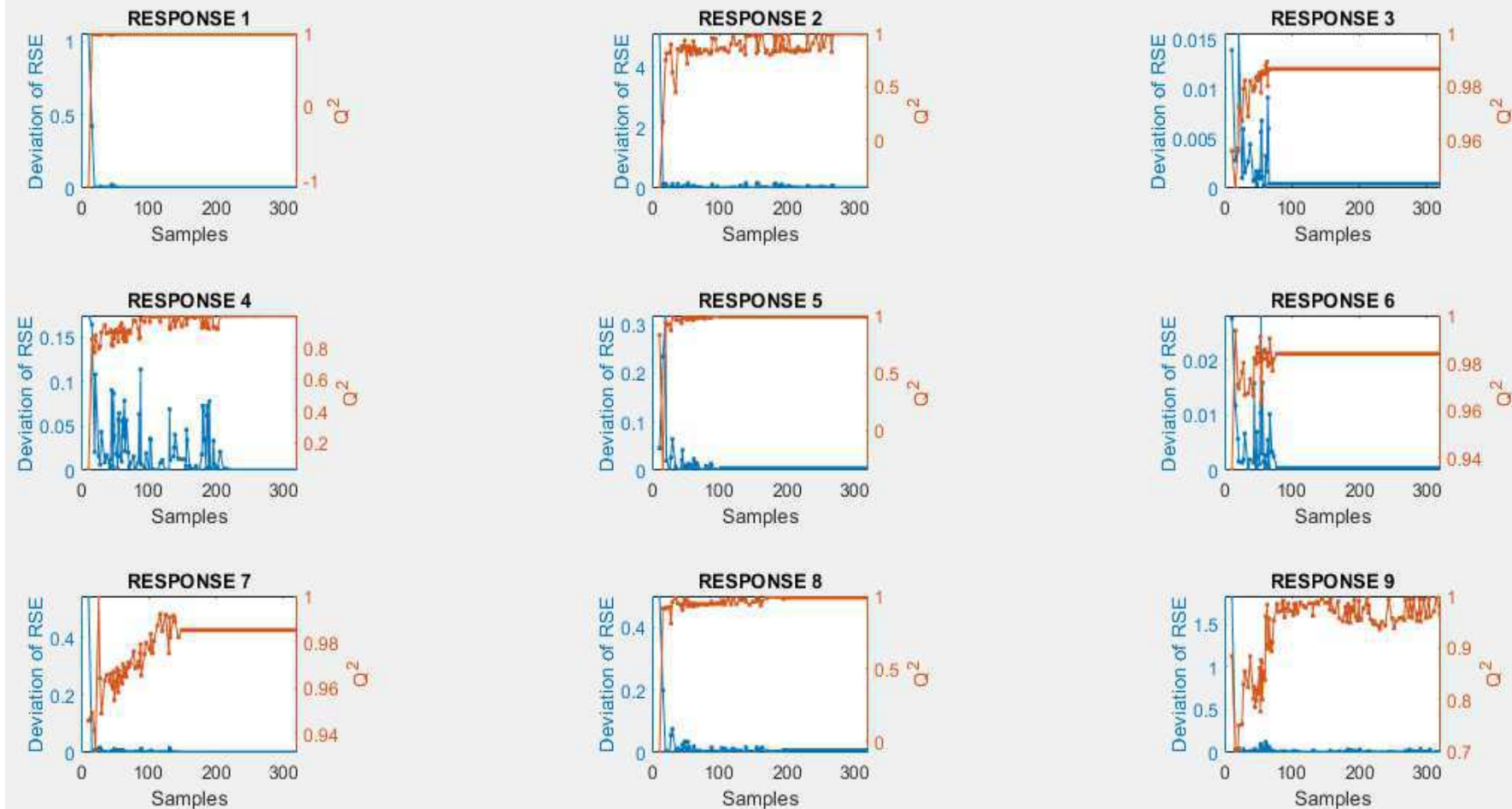


Figura 32: Evolução do processo iterativo do algoritmo aplicado à simulação de uma coluna de destilação em Aspen Plus (Caso 4). Fonte: Próprio Autor



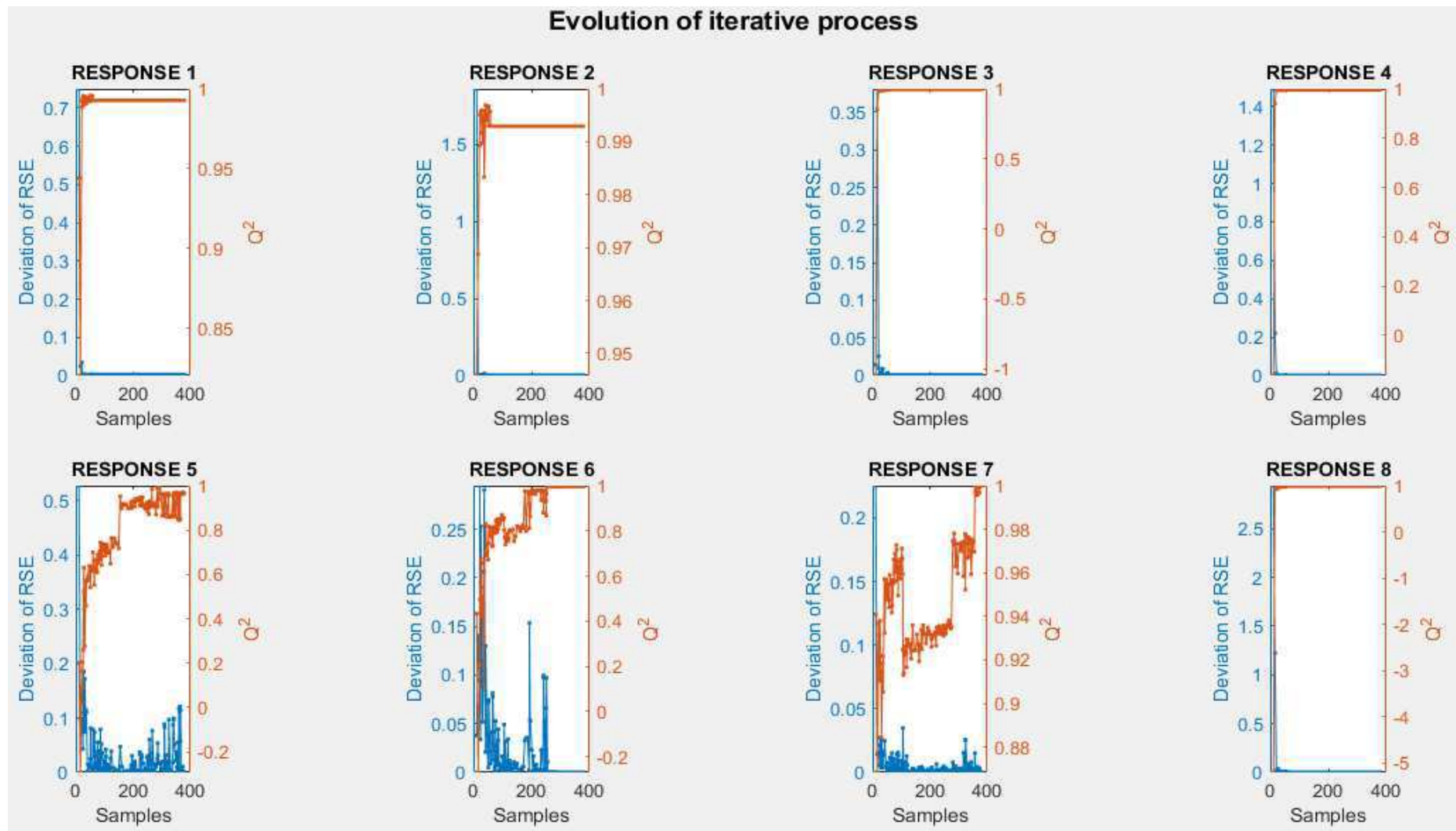


Figura 33: Evolução do processo iterativo do algoritmo aplicado à simulação de uma planta de tratamento de efluentes em Simulink (Caso 1). Fonte: Próprio Autor

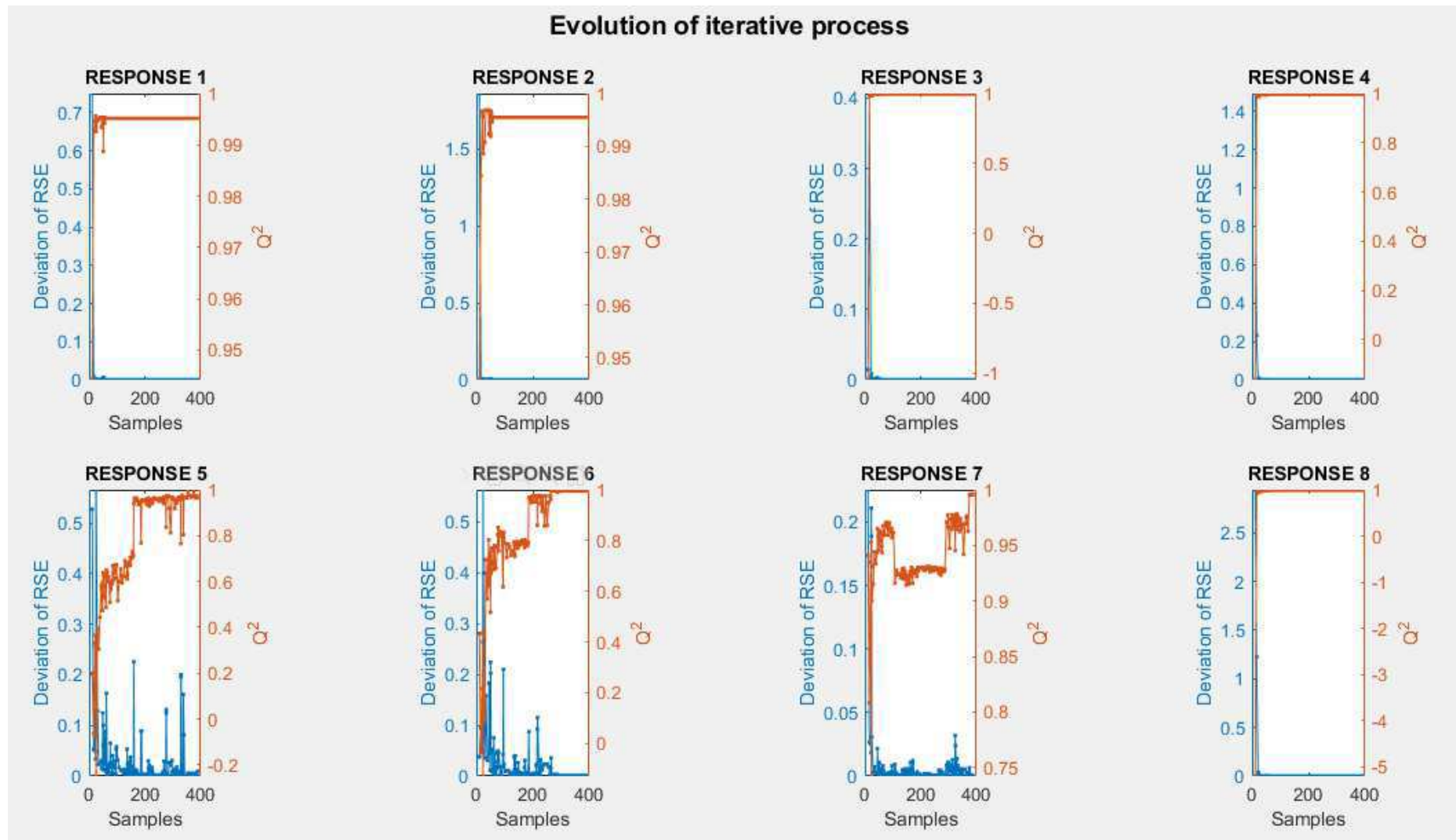


Figura 34: Evolução do processo iterativo do algoritmo aplicado à simulação de uma planta de tratamento de efluentes em Simulink (Caso 2). Fonte: Próprio Autor

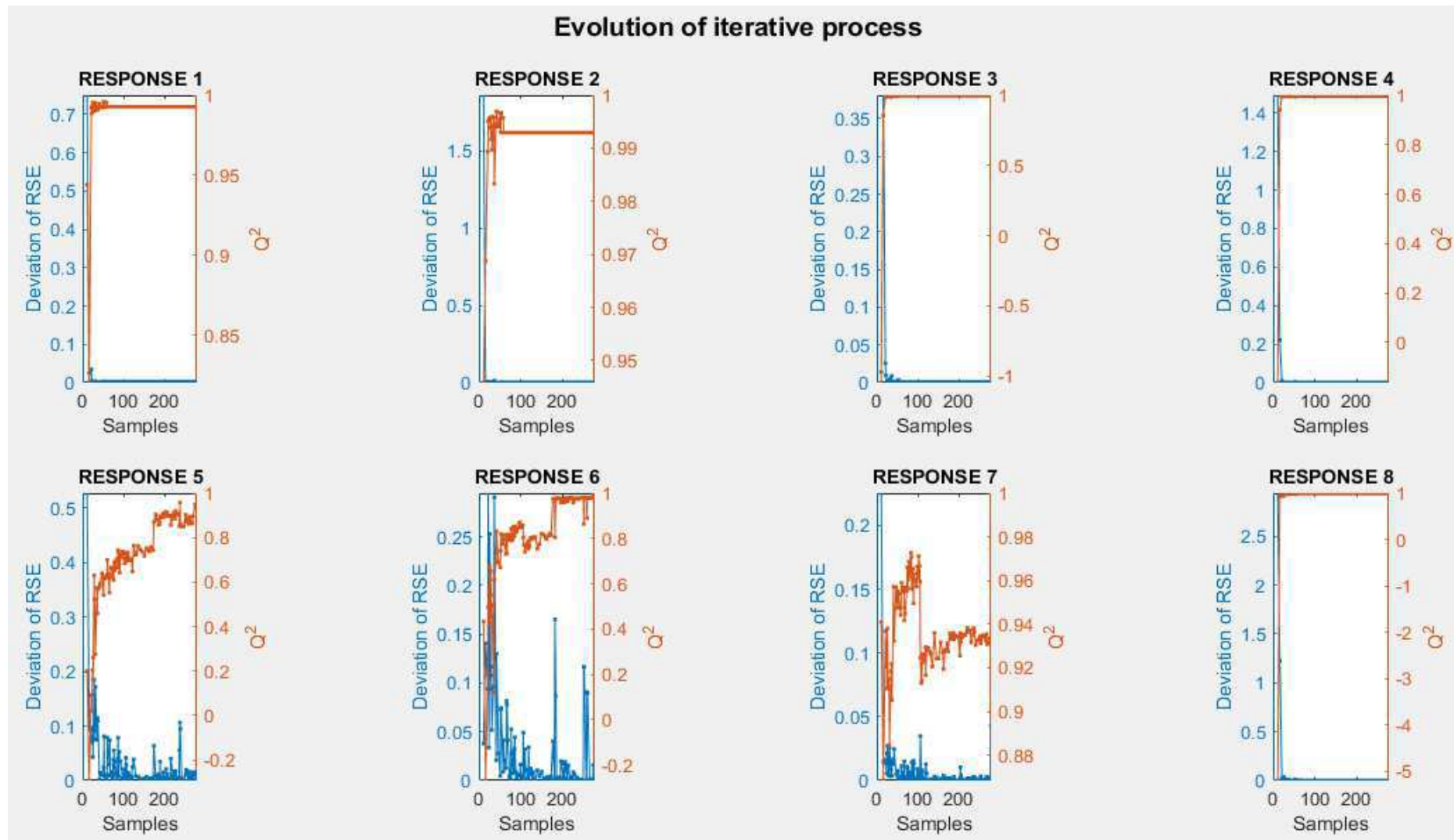
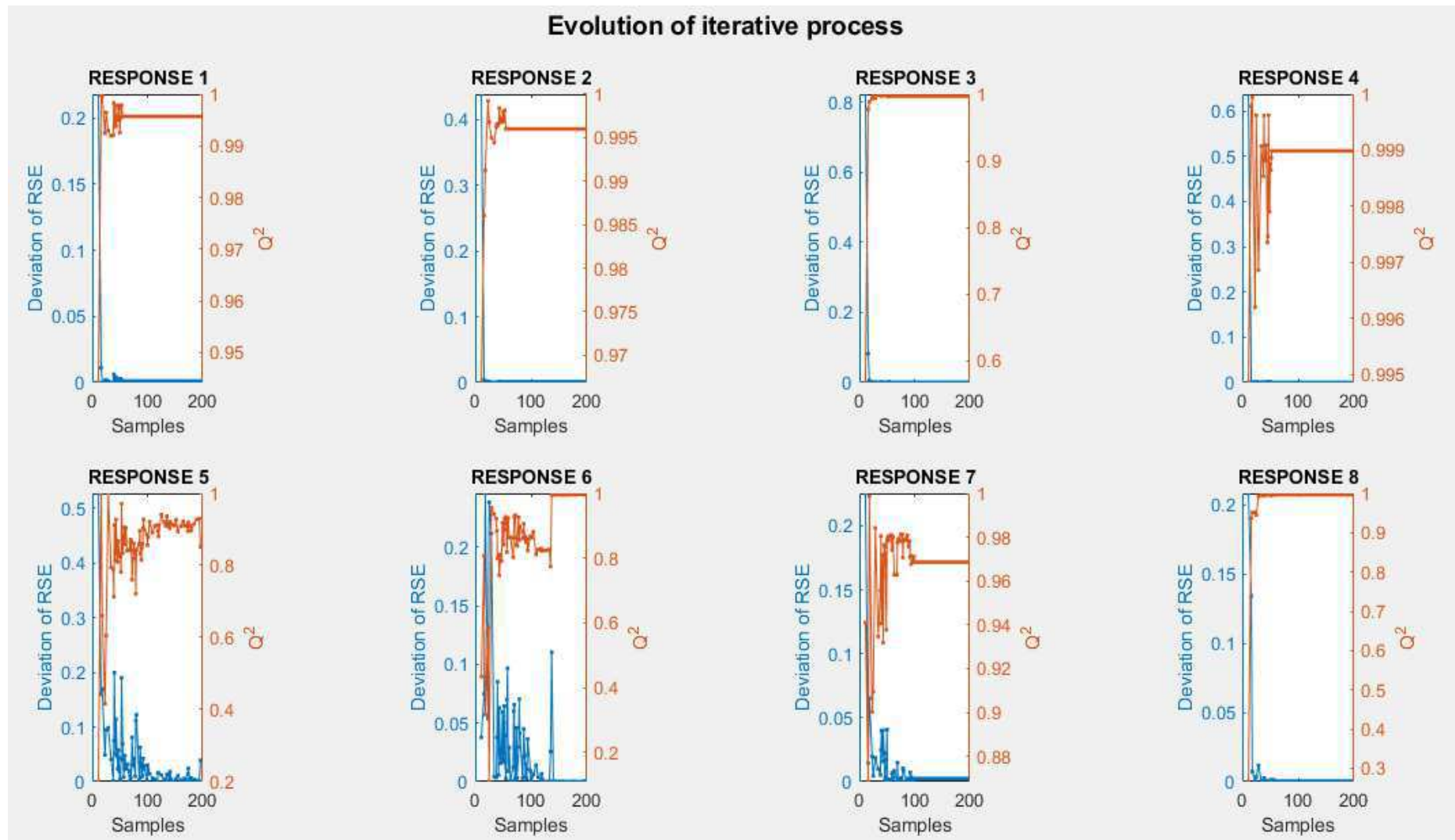


Figura 35: Evolução do processo iterativo do algoritmo aplicado à simulação de uma planta de tratamento de efluentes em Simulink (Caso 3). Fonte: Próprio Autor



**Figura 36: Evolução do processo iterativo do algoritmo aplicado à simulação de uma planta de tratamento de efluentes em Simulink (Caso 4). Fonte: Próprio Autor**

Pode-se perceber que a tendência é sempre trazer o valor do desvio do SER (Erro) ao setpoint definido “zero”, indicando que a estratégia “feedback” adotada no algoritmo realmente funciona conforme programada. Como consequência, o  $Q^2$  de validação cruzada tende a estabilizar em torno de valores altos, em todos os casos, devido ao algoritmo de otimização inteira o qual objetiva a busca do metamodelo que maximiza este valor de  $Q^2$ . No entanto, não há garantia que a convergência será atingida. Na verdade, pode-se observar que algumas respostas possuem um comportamento oscilatório expressivo, especialmente na região de valores altos de  $Q^2$ . Uma provável causa deste comportamento pode estar relacionada ao mecanismo de busca utilizado pelo algoritmo de otimização para a seleção dos metamodelos de regressão, como por exemplo, a otimização bayesiana. Contudo, um possível remédio para evitar tais oscilações seria a utilização das rotinas consagradas já mencionadas no texto (AutoWEKA, Auto-sklearn, AutoGluon ou H2O AutoML) em substituição às rotinas de otimização usadas neste trabalho para a seleção dos metamodelos de regressão.

É importante notar que a estratégia inclusiva funciona exatamente conforme programado, como mostra uma das tabelas de evolução dos conjuntos das respostas convergidas e não convergidas. Veja que respostas não convergidas passam a convergidas durante o processo iterativo graças a estratégia adotada no algoritmo que permite o processamento de respostas não-convergidas mesmo essas tendo atingido o número máximo de iterações quando estavam em processamento. Vale observar ainda que o processamento em background proposto no algoritmo inclusivo promove a convergência de várias respostas ativas durante as iterações. Isso representa uma enorme vantagem em termos de aumento de eficiência e redução de tempo computacional.

A tabela a seguir apresenta as iterações para o exemplo do Matlab (caso 2; com Lola-Voronoi).

Tabela 3: Iterações para o exemplo do Matlab (Caso 2). Fonte: Próprio Autor

| Iteração | Resposta em processamento | Conjunto de respostas ativas | Conjunto de respostas convergidas | Conjunto de respostas não convergidas |
|----------|---------------------------|------------------------------|-----------------------------------|---------------------------------------|
| 1        | 6                         | 1 2 3 4 5 6 7 8              |                                   |                                       |
| 2        | 6                         | 1 2 3 4 5 6 7 8              |                                   |                                       |
| 3        | 6                         | 1 2 3 5 6 8                  | 4 7                               |                                       |
| 4        | 6                         | 1 2 3 5 6 8                  | 4 7                               |                                       |
| 5        | 6                         | 1 2 3 5 6 8                  | 4 7                               |                                       |
| 6        | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 7        | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 8        | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 9        | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 10       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 11       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 12       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 13       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 14       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 15       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 16       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 17       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 18       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 19       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 20       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 21       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 22       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 23       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 24       | 6                         | 1 2 3 5 6                    | 4 7 8                             |                                       |
| 25       | 6                         | 1 2 3 6                      | 4 5 7 8                           |                                       |
| 26       | 6                         | 1 2 3 6                      | 4 5 7 8                           |                                       |
| 27       | 6                         | 1 3 6                        | 2 4 5 7 8                         |                                       |
| 28       | 6                         | 1 3 6                        | 2 4 5 7 8                         |                                       |
| 29       | 6                         | 1 3 6                        | 2 4 5 7 8                         |                                       |
| 30       | 6                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 31       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 32       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 33       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 34       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 35       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 36       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 37       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 38       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 39       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |
| 40       | 1                         | 1 3                          | 2 4 5 7 8                         | 6                                     |

Continuação

| <b>Iteração</b> | <b>Resposta em processamento</b> | <b>Conjunto de respostas ativas</b> | <b>Conjunto de respostas convergidas</b> | <b>Conjunto de respostas não convergidas</b> |
|-----------------|----------------------------------|-------------------------------------|--|--|
| 41              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 42              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 43              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 44              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 45              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 46              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 47              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 48              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 49              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 50              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 51              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 52              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 53              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 54              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 55              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 56              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 57              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 58              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 59              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 60              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 61              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 62              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 63              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 64              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 65              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 66              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 67              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 68              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 69              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 70              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 71              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 72              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 73              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 74              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 75              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 76              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 77              | 1                                | 1 3                                 | 2 4 5 7 8                                | 6  |
| 78              | 1                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 79              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 80              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 81              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |

Continuação

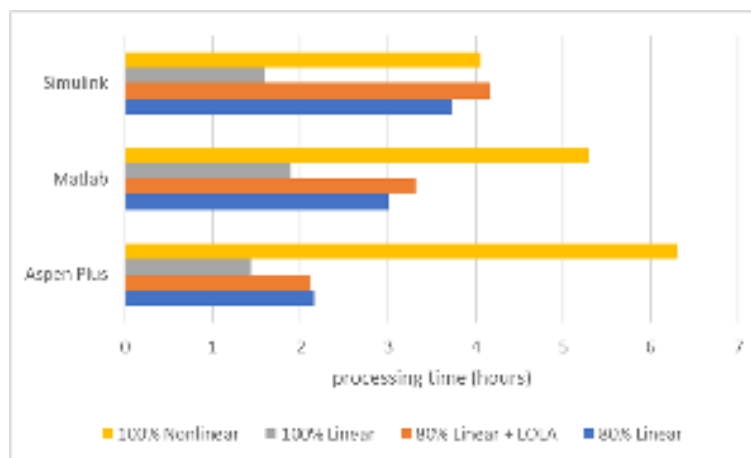
| <b>Iteração</b> | <b>Resposta em processamento</b> | <b>Conjunto de respostas ativas</b> | <b>Conjunto de respostas convergidas</b> | <b>Conjunto de respostas não convergidas</b> |
|-----------------|----------------------------------|-------------------------------------|--|--|
| 82              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 83              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 84              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 85              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 86              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 87              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 88              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 89              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 90              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 91              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 92              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 93              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 94              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 95              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 96              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 97              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 98              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 99              | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 100             | 3                                | 3                                   | 2 4 5 7 8                                | 1 6  |
| 101             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 102             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 103             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 104             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 105             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 106             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 107             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 108             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 109             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 110             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 111             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 112             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 113             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 114             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 115             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 116             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 117             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 118             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 119             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 120             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 121             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |
| 122             | 3                                | 3                                   | 1 2 4 5 7 8                              | 6  |



Continuação

| Iteração | Resposta em processamento | Conjunto de respostas ativas | Conjunto de respostas convergidas | Conjunto de respostas não convergidas |
|----------|---------------------------|------------------------------|-----------------------------------|---------------------------------------|
| 123      | 3                         | 3                            | 1 2 4 5 7 8                       | 6                                     |
| 124      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 125      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 126      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 127      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 128      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 129      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 130      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 131      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 132      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 133      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 134      | 3                         | 3                            | 1 2 4 5 6 7 8                     |                                       |
| 135      | 3                         |                              | 1 2 4 5 6 7 8                     | 3                                     |

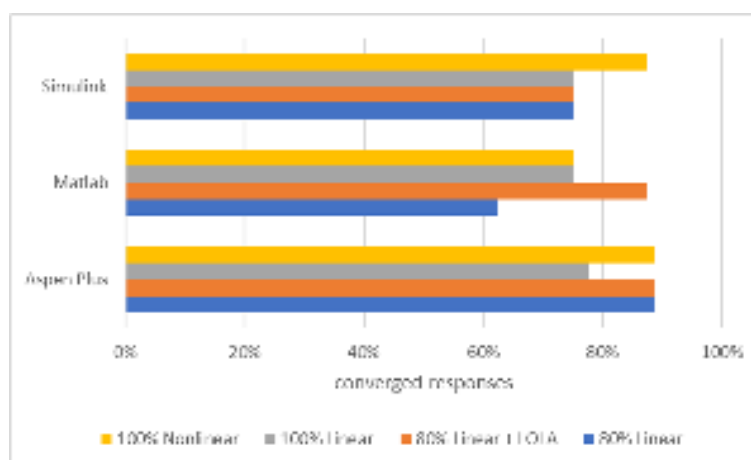
As Figuras 36 a 40 mostram os resultados das métricas de eficiência do algoritmo feedback inclusivo.



**Figura 37: Tempo de processamento do algoritmo para todos os casos estudados.**  
**Fonte: Próprio Autor**

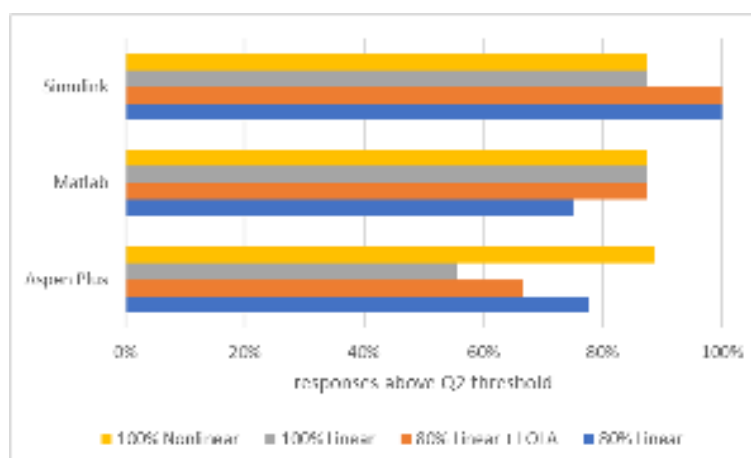
A Figura 36 apresenta o tempo de processamento do algoritmo em todos os casos estudados. Observa-se que os tempos de processamento para construção de metamodelos obtidos por métodos de regressão não-lineares são maiores do que os da regressão linear devido ao tempo gasto com a otimização dos hiper-parâmetros do modelo gaussiano, condição para obtenção de metamodelos com alta capacidade preditiva.

Os tempos de processamento podem ser reduzidos através da re-implementação do algoritmo em linguagens compiladas, como o Python, e também pelo uso de rotinas consagradas de seleção de metamodelos de regressão, como por exemplo AutoWEKA, Auto-sklearn, AutoGluon ou H2O AutoML. Entretanto, este tempo de processamento pode ser considerado irrelevante se o simulador utilizado para geração de dados levar um tempo muito mais longo para simulação das amostras. Isto é comum acontecer em simuladores de Fluidodinâmica Computacional (CFD).



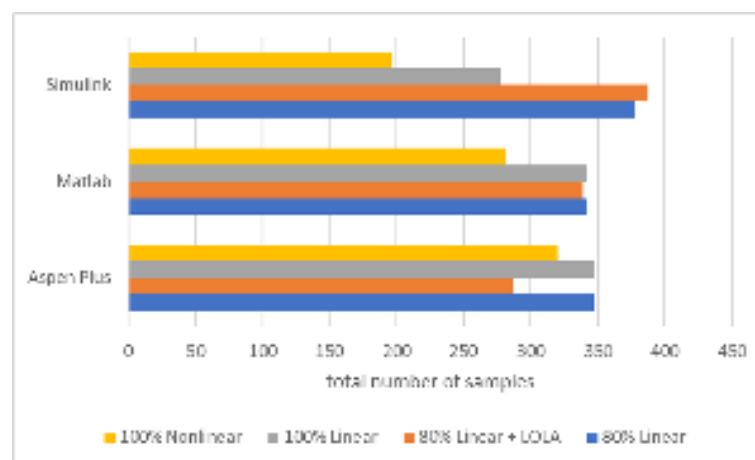
**Figura 38: Porcentagem de respostas convergidas para todos os casos estudados**

A Figura 39 apresenta a porcentagem de respostas convergidas em todos os casos estudados. Observa-se em geral que o percentual de respostas convergidas para o caso dos metamodelos obtidos puramente por métodos não-lineares foi um pouco maior, em alguns casos, em relação à regressão linear.



**Figura 39: Porcentagem de respostas com o valor do  $Q^2$  acima do limite inferior de 0,97 para todos os casos. Fonte: Próprio Autor**

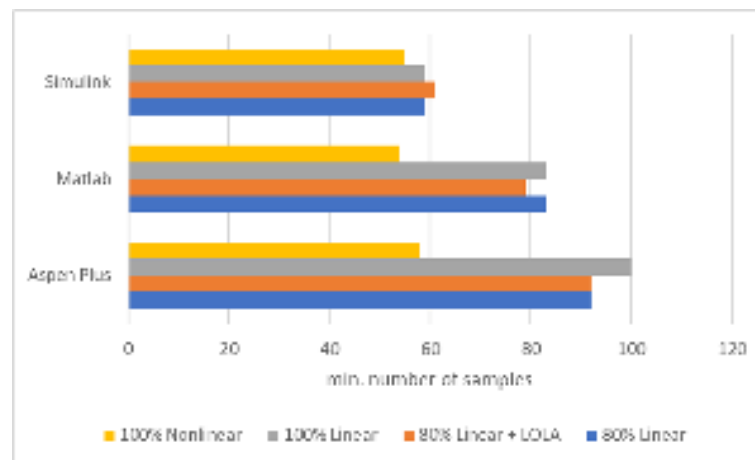
A Figura 38 apresenta a porcentagem de respostas que obtiveram o valor do  $Q^2$  acima do limite inferior de 0,97. Este limite inferior representa o padrão ideal de convergência mas esse objetivo nem sempre é atingido. Pelos resultados, observa-se o alto percentual de respostas com valor de  $Q^2$  de validação cruzada maior que o limite mínimo para o caso onde os metamodelos foram obtidos puramente por métodos não-lineares. Para os casos onde métodos lineares de geração de metamodelos foram considerados, os percentuais de  $Q^2$  de validação cruzada maior que o limite mínimo apresentaram valores significativos.



**Figura 40: Número total de amostras necessárias para a construção dos metamodelos em cada um dos casos estudados. Fonte: Próprio Autor**

A Figura 39 apresenta o número total de amostras necessárias para a construção dos metamodelos em cada caso analisado. Como esperado, metamodelos obtidos puramente por métodos não-lineares (Caso 4) requerem em geral menos amostras, apesar do tempo de processamento ser maior. Trata-se, portanto, de um caso ideal para aplicações de CFD, visto que o tempo de simulação costuma ser extremamente maior do que o tempo de processamento do algoritmo. Por outro lado, o número total de amostras usadas quando considerados os métodos de regressão linear para construção de metamodelos são maiores, isto porque o algoritmo tem que explorar uma região maior do espaço de busca na tentativa de encontrar um metamodelo que minimize o valor do Erro. Este fato ocorre devido à limitação dos metamodelos gerados por tais métodos em capturar as não-linearidades das respostas. Metamodelos oriundos de regressão linear consistem em um somatório de funções não-lineares elementares poderadas por coeficientes. Já os metamodelos oriundos de

regressão não linear possuem funções de covariância, as quais são responsáveis em ajustar as respostas a estas não-linearidades.



**Figura 41: Número mínimo de amostras necessárias para a construção dos metamodelos em cada um dos casos estudados. Fonte: Próprio Autor**

A Figura 40 apresenta o número mínimo de amostras necessárias para a construção dos metamodelos em cada um dos casos analisados. Respostas consideradas “fáceis” de convergir requerem um número menor de amostras. O algoritmo deve ser eficiente em detectar estas respostas de forma a processá-las preferencialmente em background, reduzindo o tempo de processamento e o número total de amostras. Nota-se que quando métodos puramente não-lineares são empregados, o número mínimo de amostras é sempre menor em relação aos outros casos. Isto ocorre devido a capacidade de tais métodos em convergir eficientemente respostas com baixo grau de não-linearidade. Novamente, estes métodos são ideais para aplicações em que o tempo de simulação é bem maior que o tempo de processamento, como é o caso de aplicações em CFD.

A tabela a seguir apresenta as respostas que foram processadas no algoritmo de otimização. As demais respostas de cada caso foram solucionadas em “background”.

**Tabela 4: Respostas processadas no algoritmo de otimização “kresp”. Fonte: Próprio Autor**

| <b>Simulador</b>                      | <b>Respostas</b> |
|---------------------------------------|------------------|
| Matlab (Caso 1)                       | 6,1,3            |
| Matlab (Caso 2 – Lola-Voronoi)        | 6,1,3            |
| Matlab (Caso 3 – 100% Linear)         | 6,1,3            |
| Matlab (Caso 4 – 100% não linear)     | 6,1,3            |
| Aspen Plus (Caso 1)                   | 2,9,4,1          |
| Aspen Plus (Caso 2 – Lola-Voronoi)    | 2,9,4,1          |
| Aspen Plus (Caso 3 – 100% Linear)     | 2,9,4,1          |
| Aspen Plus (Caso 4 – 100% não linear) | 1,9,8,4,2        |
| Simulink (Caso 1)                     | 8,5,6,7          |
| Simulink (Caso 2 – Lola-Voronoi)      | 8,5,6,7          |
| Simulink (Caso 3 – 100% Linear)       | 8,5,6,7          |
| Simulink (Caso 4 – 100% não linear)   | 5,6              |

Observando a Tabela 4, percebe-se que não há diferença entre a ordem das respostas “kresp” para os casos 1, 2 e 3 apresentados. Entretanto observa-se que para simulações de plantas no caso 4, a ordem de escolha da resposta “kresp” é alterada.

Em resumo, metamodelos obtidos por métodos de regressão não-linear possuem capacidade preditiva maior, convergem mais respostas (o que não caracteriza uma vantagem pois o  $Q^2$  de validação cruzada pode não exceder o limite mínimo estabelecido pelo usuário), conferem mais respostas com maiores valores do  $Q^2$  de validação cruzada (o que indica maior capacidade preditiva para um maior número de respostas), usam menos amostras e detectam eficientemente respostas com baixo grau de não-linearidade. No entanto, são os que consomem mais tempo de CPU para processamento.

Toda a análise feita nesta seção para os casos considerados revela que o uso de técnicas lineares de construção de metamodelos deveria sempre ser considerado em conjunto com um percentual do total máximo de amostras permitidas destinado às técnicas não-lineares. A sugestão é que estes percentuais dependam do tempo que o simulador leva para a obtenção dos dados para uso no algoritmo.

## 6. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho objetivou o desenvolvimento de um algoritmo “feedback” inclusivo destinado à seleção automática de metamodelos construídos a partir de dados de simulação dentro do escopo do “Automated Machine Learning”. Como resultado, foram obtidos metamodelos de estrutura simples, como os gerados pelo método de regressão linear dos mínimos quadrados com seleção de recursos, sempre que possível, com um mínimo número de amostras e no menor espaço de tempo. Desta forma, esta contribuição respondeu à questão que era relacionada ao número de amostras necessárias para construção de metamodelos. A aplicação do algoritmo aos casos representativos revelaram a eficácia da estratégia montada ao promover a construção de metamodelos com ampla capacidade preditiva. A sugestão é usar métodos de construção que gerem metamodelos não-lineares mais complexos somente em casos onde o tempo de simulação é bem maior que o tempo de processamento. Do contrário, deveria-se optar por um mix de métodos lineares e não-lineares usados no processo iterativo de construção de tais metamodelos.

Para trabalhos futuros, este trabalho apresenta as seguintes sugestões:

- Reimplementar o algoritmo em linguagem compilada, como o Python;
- Criar interface gráfica para interação com o usuário;
- Implementar comunicação do algoritmo com simuladores CFD para coleta de dados;
- Incluir comunicação do algoritmo com Aspen Hysys e Aspen Dynamics (em particular, para processos em batelada);
- Implementar “ensemble learning” no algoritmo inclusivo. O “ensemble learning” é um paradigma de aprendizado de máquina em que vários modelos são treinados para resolver o mesmo problema e combinados para obter melhores resultados. A principal hipótese é que quando os modelos “menores” são combinados corretamente pode-se obter metamodelos mais precisos ou robustos;
- Implementar o algoritmo para o caso onde os dados para regressão são fornecidos previamente pelo usuário. Neste caso, o objetivo é determinar

as melhores transformações de variáveis, o melhor escalonamento destas variáveis e o melhor metamodelo através da rotina de otimização.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, P.F.; BOURGUIGNON, B.; KHOTS, M. S.; MASSART, D.L.; PHAN-THAN-LU, R. D-Optimal designs. **Geometrics and Intelligent Laboratory Systems**. Vol. 30, pag. 199-210, 1995.

AIROLA, A.; PAHIKKALA, T.; WAEGEMAN, W.; DE BAETS, B.; SALAKOSKI, T. An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. **Computational Statistics & Data Analysis**. V. 55. Pag. 1828-1844. 2011.

AURENHAMMER, F. Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure. **ACM Computing Surveys**. vol. 23, no. 3, 1991.

BAAREH, A. K. Optimizing Software Effort Estimation Models Using Back-Propagation Versus Radial Base Function Networks. **Journal of Computer Science**. vol. 15, no. 3, pag. 321-331, 2019.

BEAUJEAN, A. A. Sample Size Determination for Regression Models Using Monte Carlo Methods in R. **Practical Assessment, Research, and Evaluation**. Vol. 12, 2014.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation Learning: A Review and New Perspectives. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. V. 35. Pag. 1798-1828. 2013.

BERRIOS, M.; GUTIÉRREZ, M. C.; MARTÍN, M. A.; MARTÍN, A. Application of the factorial design of experiments to biodiesel production from lard. **Fuel Processing Technology**. Vol. 90, pag. 1447-1451, 2009.

BHATTACHARYYA, B. A Critical Appraisal of Design of Experiments for Uncertainty Quantification. **Arch Computat Methods Eng**. Vol. 25, pag. 727-751, 2018.

BOBBIO, A.; PORTINALE, L.; MINICHINO, M.; CIANCAMERLA, E. Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. **Reliability Engineering & System Safety**. V. 71. N. 3. Pag. 249-260. 2001.

BORNN, L.; DOUCET, A.; GOTTARDO, R. An efficient computational approach for prior sensitivity analysis and cross-validation. **The Canadian Journal of Statistics**. Vol. 38, no. 1, pag. 47-64, 2010.

BOUHLEL, M. A.; BARTOLI, N.; OTSMANE, A.; MORLIER, J. Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. Vol. 53, pag. 935-352, 2016.

BOUKOUVALA, F.; FLOUDAS, C. A. ARGONAUT: Algorithms for Global Optimization of constrained grey-box computational problems. **Otim Lett**. 2016.

BOYD, S.; VANDENBERGHE, L. **Convex Optimization**. First Edition. 2004.



BROCHU, E.; CORA, V.; FREITAS, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. **Computing Research Repository**. 2010.

BUHMANN, M. D. **Radial Basis Functions (Theory and Implementations)**. 2003.

CARVALHO, F. R. D. Análise Fatorial. **Dissertação**. Universidade de Coimbra. 2013.

CELISSE, A. Optimal cross-validation in density estimation with the loss. **The Annals of Statistics**. V. 42. Pag. 1879-1910. 2014.

CHEN, H.; YANG, C.; DENG, K.; ZHOU, N.; WU, H. Multi-objective optimization of the hybrid wind/solar/fuel cell distributed generation system using Hammersley Sequence Sampling. **International Journal of Hydrogen Energy**. 2017.

CHI, H.; MASCAGNI, M.; WARNOCK, T. On the optimal Halton Sequence. **Mathematics and Computers in Simulation**. Vol. 70, pag. 9-21, 2005.

COHEN, J. Multiple regression as a general data-analytic system. **Psychological Bulletin**. Vol. 70, no. 6, pag. 426-443, 1968.

COSSIO, J. F.; DIAZ, J. F. **Maximum Entropy Method: Sampling Bias**. Disponível em: <https://arxiv.org/ftp/arxiv/papers/1507/1507.04783.pdf>. Acesso em: 20 de Ago. 2020.

COUCKUIT, I.; DECLERCQ, F.; DHAENE, T.; ROGIER, H.; KNOCKAERT, L. Surrogate-Based Infill Optimization Applied to Electromagnetic Problems. **International Journal of RF and Microwave Computer-Aided Engineering**. Vol. 20, no. 5, 2010.

COZAD, A.; SAHINIDIS, N. V.; MILLER, D. C. A combined first-principles and data-driven approach to model building. *Computers and Chemical Engineering*. N. 73. Pag. 116-127. 2015.

COZAD, A.; SAHINIDIS, N. V.; MILLER, D. C. Automatic learning of algebraic models for optimization. *AIChE Journal*. V. 60. Pag. 2211-2227. 2014.

CRESSIE, N. A. C. **Statistics for Spatial Data, Revised Edition**. Wiley-interscience. 1993.

CROMBECQ, K.; GORISSEN, D. A novel sequential design for global surrogate modeling. **Proceedings of the 2009 Winter Simulation Conference**. 2008.

CROMBECQ, K. Surrogate Modelling of Computer Experiments with Sequential Experimental Design. **Tese**. Universiteit Antwerpen. 2011.

CROMBECQ, K.; LAERMANS, E.; DHAENE, T. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. **European Journal of Operational Research**. Vol. 214, pag. 683-696, 2011.

CYBENKO. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals and Systems*. V. 2. Pag. 303-314. 1989.

DEHLENDORFF, C. **Monte Carlo Analysis**. Technical University of Denmark. 2010.

DENG, L.; YU, D. Deep Learning: Methods and Applications. **Foundations and Trends in Signal Processing**. V. 7. N. 3-4. 2014.

DIRIGNEI, D. An estimation algorithm for fast kriging surrogates of computer models with unstructured multiple outputs. **Computer Methods in Applied Mechanics and Engineering**. Vol. 321, pag. 35-45, 2017.

DRUKKER, D. M.; GATES, R. Generating Halton Sequences using Mata. **The Stata Journal**. Vol. 6, no. 2, pag. 214-228, 2006.

DU, Q.; EMELIANENKO, M. Acceleration schemes for computing centroidal Voronoi tessellations. **Numerical Linear Algebra with Applications**. Vol. 13, pag. 193-192, 2006.

DU, Q.; FABER, V.; GUNZBURGER, M. Centroidal Voronoi Tessellations: Applications and Algorithms. **Society for Industrial and Applied Mathematics**. Vol. 41, no. 4, pag. 637-676, 1999.

FANG, K.; LIN, D. K. J. Uniform Experimental Designs and their Applications in Industry. **Statistics in Industry**. Vol. 22, pag. 131-170, 2003.

FASSHAUER, G. E. **Meshfire Approximation Methods with MATLAB**. Volume 6. 2007.

FAZIO, V. S. Interpolação especial: Uma comparação analítica entre redes RBF e Krigagem. **Dissertação**. Florianópolis, SC. Universidade Federal de Santa Catarina. 2013.

FILZMOSE, P.; LIEBMANN, B.; VARMUZA, K. Repeated double cross validation. **Chemometrics**. Vol. 23, pag. 160-171, 2009.

FLAKKE, A. **uniqueToIRows & uniqueToIColumns**. 2018. Disponível em: <https://www.mathworks.com/matlabcentral/fileexchange/67118-uniqueToIRows-uniqueToIColumns>. Acesso em: 12 Set. 2021.

FRISSE, C.; SCARPEL, R. A.; FERRARI, D. B. T. P. A. Metamodelagem de Funções determinísticas por composição integrada de especialistas locais. **In: Simpósio Brasileiro de Pesquisa Operacional**. XLIII. Ubatuba. 2011.

GASPAR, B.; TEIXEIRA, A.P.; SOARES, G. C. Adaptive surrogate model with active refinement combining Kriging and a trust region method. **Reliability Engineering & System Safety**. Vol. 165, pag. 277-291, 2017.

GORISSEN, D.; COUCKUYT, I.; DEMEESTER, P.; DHAENE, T.; CROMBECQ, K. A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based

Design. **Journal of Machine Learning Research**. Vol. 11, pag. 2051-2055, 2010.

GOTTARDO, R.; RAFTERY, A. Bayesian robust transformation and variable selection: a unified approach. **The Canadian Journal of Statistics**. Vol. 37, no. 3, pag. 361-380, 2009.

GUERRA, C. F.; HANDGRAAF, J.; BAERENDS, E. J.; BICKELHAUPT, F. M. Voronoi DeformationDensity (VDD) Charges: Assessment of the Mulliken, Badern Hirshfeld, Weinhold, and VDD Methods for Charge Analysis. **Journal of Computational Chemistry**. Vol. 25, no. 2, 2003.

GUNN, S. R. **Support Vector Machine for Classification and Regression**. Faculty of Engineering, Science and Mathematics School of Eletronics and Computer Science. 1998.

GUO, S. **An Introduction to Surrogate Optimization: Intuition, Illustration, case study, and the code**. 2020. Disponível em: <https://towardsdatascience.com/an-introduction-to-surrogate-optimization-intuition-illustration-case-study-and-the-code-5d9364aed51b>. Acesso em: 27 Jan 2021.

GURRIERI, S. An analysis of Sobol Sequence and the Brownian Bridge. **SSRN Electronic Journal**. 2011.

GUTMANN, H. M. A Radiad Basis Function Method for Global Optimization. **Journal of Global Optimization**. Vol. 19, pag. 201-227, 2001.

HERNANDES, A. S.; LUCAS, T. W.; CARLYLE, M. Constructing Nearly Orthogonal Latin Hypercubes for Any Nonsaturated Run-Variable Combination. **ACM Transactions on Modeling and Computer Simulation**. Vol. 22, no. 4. 2012.

HIPP, J.; GÜNTZER, U.; NAKHAEIZADEH, G. Algorithms for association rule mining – A general survey and comparison. **SIGKDD Explorations**. V. 2. Pag. 1-58. 2000.

HOCKING, R. R. A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear Regression. **Biometrics**. Vol. 32, no. 1, pag. 1-49, 1976.

JENKINS, D. G.; QUINTANA-ASCENCIO, P. F. A Solution to minimum sample size for regressions. **Public Library Science**. Vol. 15, no. 2, 2020.

JEPSSON, U.; NOPENS, I.; BENNEDETTI, L.; PONS, M. N.; ALEX, J.; COOP, J.; GERNAEY, K. V.; ROSEN, C.; SETEYER, J. P.; VANROLLEGHEM, P. Benchmark simulation model No2. **Department of Industrial Electrical Engineering and Automation**. Lund University. 2011.

JOE, S.; FUO, F. Y. Remark on Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator. **ACM Transactions on Latemactical Software**. Vol. 29, no. 1, pag. 49-57, 2003.

JOHNSON, M. E.; MOORE, L. M.; YLVISAKER, D. Minimax and maximin distance designs. **Journal of Statistical and Inference**. Vol. 26, pag. 131-148, 1990.

JONES, D.; SCHONLAU, M.; WELCH, W. Efficient global optimization of expensive black box functions. **Journal of Global Optimization**. V. 13. Pag. 455-492. 1998.

KE, L.; QIU, H.; CHEN, Z.; CHI, L. Engineering Design Based on Hammersley Sequences Sampling Method and SVR. **Advanced Materials Research**. Vol. 544, pag. 206-211, 2012.

KICSINY, R. Improved multiple linear regression based models for solar collectors. **Renewable Energy**. Vol. 91, pag. 224-232, 2016.

KIM, L.; LOH, H. Classification trees and bivariate linear discriminant node models. **American Statistical Association**. Vol. 12, no. 3, pag. 512-530, 2003.

KOCIS, L.; WHITEN, W. J. Computational investigations of low-discrepancy sequences. **ACM Transactions on Mathematical Software**. Vol. 23, no. 2, pag. 266-294, 1997.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. **Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence**. San Mateo. V. 2. Pag. 1137-1143. 1995.

KORB, K. B.; NICHOLSON, K. E. **Bayesian Artificial Intelligence**. Chapman & Hall. 2003.

KOTTHOFF, L.; THORNTON, C.; HOOS, H. H.; HUTTER, F.; LEYTON-BROWN, K. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Machine Learning Research*. N. 17. Pag. 1-5. 2016.

KRIGE, D. G. A statistical approach to some basic mine valuation problems on the Witwatersrand. **Journal of the Chemical, Metallurgical and Mining Society of South Africa**. Pag. 52-139, 1951.

KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. New York. Springer. 2013.

KUSHNER, H. J. A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise. **Basic Engineering**. V. 86. Pag. 97-106-1964.

LAAN, M.; POLLEY, E.; HUBBARD, A. Statistical applications in genetics and molecular biology. **Super learner**. V.6. 2007.

LENDELL, E.; POIRIER, S. H2O AutoML: Scalable Automatic Machine Learning. **Workshop on Automated Machine Learning**. N. 7. 2020.

LIANG, H.; SONG, W. Improved estimation in multiple linear regression models with measurement error and general constraint. **Journal of Multivariate Analysis**. Vol. 100, pag. 726-741, 2009.

LINDEN, R. **Algoritmos genéticos – uma importante ferramenta da inteligência computacional**. Edição 2. 2008.

LIU, Z.; ZHAN, J.; TAN, C. Improved Reliability Approximate Method Combining Kriging and Importance Sampling. **In: Prognostics & System Health Management Conference**. 2012.

LIZOTTE, D. Practical Bayesian Optimization. **Tese**. University of Alberta. Edmonton. Alberta. Canadá. 2008.

LOPHAVEN, S. N.; NIELSEN, H. B.; SONDEGAARD, J. Aspects of Matlab Toolbox DACE. **Technical University of Denmark**. 2002.

MANNARSWAMY, A.; MUNSON-MCGEE, S. H.; STEINER, R.; ANDERSEN, P. K. D-optimal experimental designs for Freudlich and Lagmuir adsorption isotherms. **Chemometrics and Intelligent Laboratory Systems**. Vol. 97, pag. 146-151, 2009.

MARTIN, J. D.; SIMPSON, T. W. A study on the use of kriging models to approximate deterministic computer models. **In: Design Engineering Technical Conferences and Computers**. Chicago. 2003.

MATHERON, G. The Theory of Regionalized Variables and its Applications. **Les Cahiers du Centre de Morphologie Mathématique de Fontainebleau**. No. 5. Fontenebleau. 1971.

MATHWORKS. **1-D data interpolation (table lookup) – MATLAB interp1**. 2020. Disponível em: <https://www.mathworks.com/help/matlab/ref/interp1.html>. Acesso em: 12 Set 2021.

MATHWORKS. **Mastering Machine Learning**. 2019.

MATT, J. **Extract linearly independent subset of matrix columns**. 2020. Disponível em: <https://www.mathworks.com/matlabcentral/fileexchange/77437-extract-linearly-independent-subset-of-matrix-columns>. Acesso em: 25 Out 2021.

MCLACHLAN, G.; DO, K.; AMBROISE, C. **Analyzing microarray gene expression data**. Wiley. 2004.

MENZIES, T.; HU, Y. Data Mining For Busy People. **IEEE Computer**. Pag 18-25. 2003.

MOLINARO, A. M.; SIMON, R.; PFEIFFER, R. M. Prediction error estimation: a comparison of resampling methods. **Bioinformatics**. V. 21. Pag. 3301-3307. 2005.

MOLNAR, C. **Interpretable Machine Learning**. A Guide for Making Black Box Models Explainable. 2021.

MONESS, E.; LINSLEY, M. J.; GARZON, I. E. Comparing different fractions of a factorial design: A metal cutting case study. **Applied Stochastic Models in Business and Industry**. Vol. 23, pag. 117-128, 2007.

MONTGOMERY, D. C. **Design and Analysis of Experiments**. Wiley. 2017.

MORDECAI, A. **Nonlinear Programming: Analysis and Methods**. Dover Publications. 2012.

NGUYEN, P. H. Estimating configurational entropy of complex molecules: A novel variable transformation approach. **Chemical Physics Letters**. Vol. 468, pag. 90-93, 2009.

OKABE, A.; BOOTS, B.; SUGIHARA, K. **Spatial Tessellations, Concepts and Applications of Voronoi Diagrams**. Wiley. 1992.

OISSON, A.; SANDBERG, G.; DAHLBLOM, O. On Latin hypercube sampling for structural reliability analysis. **Structural Safety**. Vol. 25, no. 1, pag. 47-68, 2003.

PLOTKIN, G. D. Automatic Methods of Inductive Inference. **Tese**. University of Edinburgh. 1970.

POWELL, M. J. D. **The Theory of Radial Basis Function Approximation**. Vol. 2, 1990.

RAMIREZ, A. B.; ANTONIO, C. G. Multiobjective Optimization of Chemical Processes with Complete Models using MATLAB and Aspen Plus. **Computacion y Sistemas**. V. 22. 2018.

RASMUSSEN, C. E.; NICKISH, H. Gaussian Processes for Machine Learning (GPML) Toolbox. **Journal of Machine Learning Research**. V. 11. Pag. 3011-3015. 2010.

RASMUSSEN, C. E.; WILLIAMS, C. K. I. **Gaussian Processes for Machine Learning**. MIT Press. 2006.

RASMUSSEN, C. E.; NICKISH, H. **The GPML Toolbox version 4.2**. Manual. 2018.

REGIS, R. G. Trust regions in Kriging-based optimization with expected improvement. **Engineering Optimization**. Pag. 1-23, 2015.

REGIS, R. G.; SHOEMAKER, C. A. A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions. **INFORMS Journal on Computing**. Vol. 19, no. 4, pag. 497-509, 2007.

RILEY, R. D.; SNELL, K. I. E.; ENSOR, J.; BURKE, D. L.; HARRELL JR, F. E.; MOONS, K. G. M.; COLLINS, G. S. Minimum sample size for developing a multivariable prediction model: Part I - Continuous outcomes. **Statistics in Medicine**. Pag. 1-14, 2018.

ROMERO, D. A.; MARIN, V. E.; AMON, C. H. Error Metrics and the Sequential Refinement of Kriging Metamodels. **Journal of Mechanical Design**. Vol. 137, no. 1, 2015.

RYU, J. S.; KIM, M. S.; CHA, K. J.; LEE, T. H.; CHOI, D. Kriging interpolation methods in geostatistics and DACE model. **Journal of Mechanical Science and Technology**. Vol. 16, no. 5, pag. 619-632, 2002.

SACKS, J.; WELCH, W. J.; MITCHELL, T. J.; WYNN, H. P. Design and Analysis of Computer Experiments. **Institute of Mathematical Statistics**. Vol. 4, no. 4, pag. 409-423, 1989.

SAHIDINIS, N. V.; MILLER, D. C. Learning Surrogate Models for Simulation-Based Optimization. **AIChE Journal**. Vol. 60, no. 6, 2014.

SALLABERRY, C. J.; HELTON, J. C.; HORA, S. C. Extension of Latin hypercube samples with correlated variables. **Reliability Engineering & System Safety**. Vol. 93, pag. 1047-1059, 2008.

SANTOS, K. R. M. Técnicas de amostragem inteligente em simulação Monte Carlo. **Dissertação**. São Carlos. Universidade de São Paulo. 2014.

SCHLIER, C. On scrambled Halton sequences. **Applied Numerical Mathematics**. Vol. 58, pag. 1467-1478, 2008.

SEADER, J. D.; HENLEY, E. J.; ROPER, D. K. **Separation Process Principles. Chemical and Biochemical Operations**. Terceira Edição. John Wiley & Sons. 2011.

SEBASTIANI, P.; WYNN, H. P. Maximum entropy sampling and optimal Bayesian experimental design. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**. Vol. 62, no. 1, pag. 145-457, 2000.

SEGARAN, T. **Programming Collective Intelligence**. First Edition. 2007.

SHAH, H. A Full Overview of Artificial Neural Network (ANN). 2020. Disponível em: <https://learn.g2.com/artificial-neural-network>. Acesso em: 26 Maio 2021.

SHAHSAVANI, D.; GRIMVALL, A. Variance-based sensitivity analysis of model outputs using surrogate models. **Environmental Modelling & Software**. Vol. 26, pag. 723-730, 2011.

SIAUW, T.; BAYEN, A. M. **An introduction to MATLAB programming and numerical methods for engineers**. Academic Press. 2014.

SINHA, S. K.; XU, X. Sequential D-optimal designs for generalized linear mixed models. **Journal of Statistical Planning and Inference**. Vol. 141, pag. 1394-1402, 2011.

SMITH, K. **Biometrika**. 1918.

SNOEK, J.; LAROCHELLE, H.; ADAMS, R. **Practical Bayesian optimization of machine learning algorithms**. V.1. 2012.

SOBOL, I. M. On the distribution of points in a cube and the approximate evaluation of integrals. **USSR Computational Mathematics and Mathematical Physics**. Vol. 7, no. 4, pag. 86-112, 1967.

SUMOWIKI. **SED: SED toolbox**. Disponível em: [http://sumowiki.intec.ugent.be/SED:SED\\_toolbox#Rules\\_of\\_thumb\\_for\\_selecting\\_the\\_right\\_sequential\\_design\\_method](http://sumowiki.intec.ugent.be/SED:SED_toolbox#Rules_of_thumb_for_selecting_the_right_sequential_design_method). Acesso em: 10 de Ago. 2020.

SUPPORT MINITAB. **Transformações de variáveis de resposta**. Disponível em: <https://support.minitab.com/pt-br/minitab/18/help-and-how-to/modeling-statistics/regression/supporting-topics/model-assumptions/transformations-of-response-variables/>. Acesso em: 8 de Set. 2020.

SURJANOVIC, S.; BINGHAM, D. **Virtual library of Simulation Experiments**. Simon Fraser University. 2013.

TAHERDOOST, H. Sampling Methods in Research Methodology; How to Choose a Sampling Technique for Research. **International Journal of Academic Research in Management**. Vol. 5, no. 2, pag. 18-27, 2016.

TEZUKA, S. A Generalization of Faure Sequences and its Efficient Implementation. **Computer Science**. 1994.

THORNTON, C.; HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. **SIGKDD international conference on Knowledge discovery and data mining**. Pag. 847-855. 2013.

VIANA, F. A. C.; VENTER, G.; BALANOV, V. An algorithm for fast optimal Latin hypercube design of experiments. **International Journal for Numerical Methods in Engineering**. Vol. 82, pag. 135-156, 2010.

VICARIO, G.; CRAPAROTTA, G.; POSTONE, G. Meta-models in Computer Experiments: Kriging versus Artificial Neural Networks. **Quality and Reliability Engineering International**. 2016.

WALPOLE, R. E.; MYERS, R. H.; MYERS, S. L.; YE, K. **Probability & Statistics for Engineers & Scientists**. Ninth Edition. 2011.

WANG, Q. J. A bayesian method for multi-site stochastic data generation: Dealing with non-concurrent and missing data, variable transformation and parameter uncertainty. **Environmental Modelling & Software**. Vol. 23, pag. 412-421, 2008.

WANG, Y.; MYERS, R. H.; SMITH, E. P.; YE, K. D-optimal designs for Poisson regression models. **Journal of statistical planning and inference**. Vol. 136, pag. 2831-2845, 2006.



WANG, S.; MCCRMICK, T. H. LEEK, J. T. Methods for correcting inference based on outcomes predicted by machine learning. **Proceedings of the National Academy of Sciences**. 2020.

WANG, J. T.; WANG, C. J.; ZHAO, J. P. Frequency response function-based model updating using Kriging model. **Mechanical Systems and Signal Processing**. 2017.

WAN, X.; PEKNY, J. F.; REKLAITIS, G. V. Simulation-based optimization with surrogate models – Application to supply chain management. **Computers & Chemical Engineering**. Vol. 19, pag. 1317-1328, 2005.

WERNICK, M. N.; YANG, Y.; BRANKOV, J. G.; YOURGANOV, G. Machine Learning in Medical Imaging. **IEEE Signal Processing Magazine**. V. 27. Pag. 25-38. 2010.

WILSON, Z. T.; SAHIDINIS, N. V. The Alamo approach to machine learning. **Computers & Chemical Engineering**. 2017.

XU, Q.; LIANG, Y.; DU, Y. Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration. **Journal of Chemometrics**. Vol. 18, pag. 112-120, 2004.

YAHIAOUI, I.; AISSANI-BENISSAD, D.; AIT-AMAR, H. Optimization of Silver Cementations Yield in Fixed Bed Reactor using Factorial Design and Central Composite Design. **The Canadian Journal of Chemical Engineering**. Vol. 88, 2010.

YAMAMOTO, J.; CONDE, R. Classificação de Recursos Minerais usando a variância de interpolação. **Revista Brasileira de Geociências**. Vol. 29, pag. 349-356, 1999.

YIN, R. K. **Case study reseach, design and methods**. 2003.

YU, J.; GOOS, P.; VANDERBROEK, M. Comparing different sampling schemes for approximating the integrals involved in the efficient design of stated choice experiments. **Transportation Research Part B**. Vol. 44, pag. 1268-1289, 2010.

ZHANG, Y.; CHANYOUNG, P.; NAM, H.; HAFTKA, R. T. Function Prediction at One Inaccessible Point Using Converging Lines. **Journal of Mechanical Design**. Vol. 139, no. 5, 2017.

ZHANG, H.; WANG, Y. Kriging and cross-validation for massive spatial data. **Environmetrics**. Vol. 21, pag. 290-304, 2010.

Zhou, J. D-optimal minimax regression designs on discrete design space. **Journal of Statistical planning and inference**. Vol. 138, pag. 4081-4092, 2008.

ZHOU, J.; TURNG, L. Process Optimization of Injection Molding Using an Adaptive Surrogate Model With Gaussian Process Approach. **Polymer Engineering and Science**. 2007

## APÊNDICE A.

Sumários das informações do processo de construção de todos os metamodelos.

**Tabela 5: Matlab Caso1**

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                  | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|--|--------------------|
| 1     | 'Unconverged' | 'no'                | 0.932512877295056                    | 'GPR with "squarexponential" covariance function'    | 342                |
| 2     | 'Converged'   | 'no'                | 0.994739419549278                    | 'OLS with feature selection'                         | 142                |
| 3     | 'Unconverged' | 'no'                | 0.999967327997049                    | 'GPR with "matern32" covariance function'            | 342                |
| 4     | 'Converged'   | 'yes'               | 1                                    | 'OLS with feature selection'                         | 83                 |
| 5     | 'Converged'   | 'no'                | 0.991766325963268                    | 'OLS with feature selection'                         | 196                |
| 6     | 'Unconverged' | 'no'                | 0.926585693827701                    | 'GPR with "ardperiodicmatern52" covariance function' | 342                |
| 7     | 'Converged'   | 'yes'               | 1                                    | 'OLS with feature selection'                         | 84                 |
| 8     | 'Converged'   | 'yes'               | 0.999999120437316                    | 'OLS with feature selection'                         | 114                |

**Tabela 6: Matlab Caso 2 (Lola-Voronoi)**

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão   | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|---|--------------------|
| 1     | 'Converged'   | 'yes'               | 0.913802490116617                    | 'GPR with "squarexponential" covariance function'             | 288                |
| 2     | 'Converged'   | 'yes'               | 0.983494596948956                    | 'OLS with feature selection'                                  | 128                |
| 3     | 'Unconverged' | 'no'                | 0.987037670062193                    | 'GPR with "ardperiodicrationalquadratic" covariance function' | 339                |
| 4     | 'Converged'   | 'yes'               | 1                                    | 'OLS with feature selection'                                  | 79                 |
| 5     | 'Converged'   | 'no'                | 0.989342472603396                    | 'OLS with feature selection'                                  | 125                |
| 6     | 'Converged'   | 'no'                | 0.995015632384107                    | 'GPR with "squarexponential" covariance function'             | 320                |
| 7     | 'Converged'   | 'yes'               | 1                                    | 'OLS with feature selection'                                  | 79                 |
| 8     | 'Converged'   | 'yes'               | 0.999999967438788                    | 'OLS with feature selection'                                  | 84                 |

Tabela 7: Matlab Caso 3 (100% Linear)

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                   | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|---|--------------------|
| 1     | 'Unconverged' | 'no'                | 0.928714544553013                    | 'GPR with "ardrationalquadratic" covariance function' | 342                |
| 2     | 'Converged'   | 'no'                | 0.994557741310222                    | 'OLS with feature selection'                          | 209                |
| 3     | 'Unconverged' | 'no'                | 0.990205114569002                    | 'OLS with feature selection'                          | 342                |
| 4     | 'Converged'   | 'yes'               | 1                                    | 'OLS with feature selection'                          | 83                 |
| 5     | 'Converged'   | 'no'                | 0.991191997194272                    | 'OLS with feature selection'                          | 180                |
| 6     | 'Converged'   | 'yes'               | 0.975187139027213                    | 'GPR with "squarexponential" covariance function'     | 187                |
| 7     | 'Converged'   | 'yes'               | 1                                    | 'OLS with feature selection'                          | 84                 |
| 8     | 'Converged'   | 'yes'               | 0.999999836224329                    | 'OLS with feature selection'                          | 116                |

Tabela 8: Matlab Caso 4 (100% não linear)

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                  | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|--|--------------------|
| 1     | 'Unconverged' | 'no'                | 0.943366547443433                    | 'GPR with "ardgabor" covariance function'            | 282                |
| 2     | 'Converged'   | 'yes'               | 0.995845886880569                    | 'GPR with "squarexponential" covariance function'    | 168                |
| 3     | 'Converged'   | 'yes'               | 0.999987772102903                    | 'GPR with "ardperiodicmatern52" covariance function' | 282                |
| 4     | 'Converged'   | 'yes'               | 1                                    | 'OLS with feature selection'                         | 54                 |
| 5     | 'Converged'   | 'no'                | 0.983964205683187                    | 'GPR with "squarexponential" covariance function'    | 116                |
| 6     | 'Unconverged' | 'no'                | 0.996321810199090                    | 'GPR with "ardperiodicmatern52" covariance function' | 282                |
| 7     | 'Converged'   | 'yes'               | 1                                    | 'OLS with feature selection'                         | 55                 |
| 8     | 'Converged'   | 'yes'               | 0.99999999729212                     | 'GPR with "squarexponential" covariance function'    | 111                |

Tabela 9: Aspen Plus Caso 1

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                  | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|--|--------------------|
| 1     | 'Converged'   | 'yes'               | 0.957869873240457                    | 'OLS with feature selection'                         | 347                |
| 2     | 'Converged'   | 'yes'               | 0.987616346034445                    | 'GPR with "ardperiodicmatern52" covariance function' | 141                |
| 3     | 'Converged'   | 'yes'               | 0.985798880161756                    | 'OLS with feature selection'                         | 92                 |
| 4     | 'Unconverged' | 'no'                | 0.999097449905682                    | 'GPR with "squaredexponential" covariance function'  | 347                |
| 5     | 'Converged'   | 'yes'               | 0.985653917288413                    | 'OLS with feature selection'                         | 139                |
| 6     | 'Converged'   | 'yes'               | 0.983303359378001                    | 'OLS with feature selection'                         | 101                |
| 7     | 'Converged'   | 'yes'               | 0.981165262064719                    | 'OLS with feature selection'                         | 136                |
| 8     | 'Converged'   | 'yes'               | 0.949898748051213                    | 'OLS with feature selection'                         | 160                |
| 9     | 'Converged'   | 'yes'               | 0.990933017941704                    | 'GPR with "ardperiodicmatern52" covariance function' | 301                |

Tabela 10: Aspen Plus Caso 2 (Lola-Voronoi)

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                  | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|--|--------------------|
| 1     | 'Converged'   | 'yes'               | 0.957347559961815                    | 'OLS with feature selection'                         | 287                |
| 2     | 'Converged'   | 'no'                | 0.991273959687103                    | 'GPR with "ardperiodicmatern52" covariance function' | 147                |
| 3     | 'Converged'   | 'yes'               | 0.984124150283802                    | 'OLS with feature selection'                         | 93                 |
| 4     | 'Unconverged' | 'no'                | 0.998220445651789                    | 'GPR with "squaredexponential" covariance function'  | 287                |
| 5     | 'Converged'   | 'yes'               | 0.985782530313727                    | 'OLS with feature selection'                         | 95                 |
| 6     | 'Converged'   | 'yes'               | 0.981615609257594                    | 'OLS with feature selection'                         | 114                |
| 7     | 'Converged'   | 'yes'               | 0.975677164922070                    | 'OLS with feature selection'                         | 92                 |
| 8     | 'Converged'   | 'yes'               | 0.940099263239464                    | 'OLS with feature selection'                         | 165                |
| 9     | 'Converged'   | 'yes'               | 0.914864742150115                    | 'OLS with feature selection'                         | 174                |

Tabela 11: Aspen Plus Caso 3 (100% Linear)

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                  | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|--|--------------------|
| 1     | 'Converged'   | 'yes'               | 0.957347559961815                    | 'OLS with feature selection'                         | 287                |
| 2     | 'Converged'   | 'no'                | 0.991273959687103                    | 'GPR with "ardperiodicmatern52" covariance function' | 147                |
| 3     | 'Converged'   | 'yes'               | 0.984124150283802                    | 'OLS with feature selection'                         | 93                 |
| 4     | 'Unconverged' | 'no'                | 0.998220445651789                    | 'GPR with "squaredexponential" covariance function'  | 287                |
| 5     | 'Converged'   | 'yes'               | 0.985782530313727                    | 'OLS with feature selection'                         | 95                 |
| 6     | 'Converged'   | 'yes'               | 0.981615609257594                    | 'OLS with feature selection'                         | 114                |
| 7     | 'Converged'   | 'yes'               | 0.975677164922070                    | 'OLS with feature selection'                         | 92                 |
| 8     | 'Converged'   | 'yes'               | 0.940099263239464                    | 'OLS with feature selection'                         | 165                |
| 9     | 'Converged'   | 'yes'               | 0.914864742150115                    | 'OLS with feature selection'                         | 174                |

Tabela 12: Aspen Plus Caso 4 (100% Não linear)

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                    | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|--|--------------------|
| 1     | 'Converged'   | 'no'                | 0.989708571829076                    | 'GPR with "squaredexponential" covariance function'    | 58                 |
| 2     | 'Converged'   | 'yes'               | 0.996062853150310                    | 'GPR with "matern32" covariance function'              | 320                |
| 3     | 'Converged'   | 'yes'               | 0.986626296250506                    | 'OLS with feature selection'                           | 66                 |
| 4     | 'Converged'   | 'yes'               | 0.998960134959784                    | 'GPR with "ardgabor" covariance function'              | 264                |
| 5     | 'Converged'   | 'no'                | 0.989020575578295                    | 'OLS with feature selection'                           | 102                |
| 6     | 'Converged'   | 'yes'               | 0.983877217168739                    | 'GPR with "squaredexponential" covariance function'    | 78                 |
| 7     | 'Converged'   | 'yes'               | 0.985102684642280                    | 'GPR with "squaredexponential" covariance function'    | 149                |
| 8     | 'Converged'   | 'yes'               | 0.989048104089269                    | 'GPR with "ardsquaredexponential" covariance function' | 200                |
| 9     | 'Unconverged' | 'no'                | 0.959863189298849                    | 'GPR with "ardperiodicmatern52" covariance function'   | 320                |

Tabela 13: Simulink Caso 1

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                    | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|--|--------------------|
| 1     | 'Converged'   | 'yes'               | 0.992779911717020                    | 'OLS with feature selection'                           | 59                 |
| 2     | 'Converged'   | 'yes'               | 0.992962524239706                    | 'OLS with feature selection'                           | 59                 |
| 3     | 'Converged'   | 'yes'               | 0.995300800493081                    | 'OLS with feature selection'                           | 59                 |
| 4     | 'Converged'   | 'yes'               | 0.995816221783398                    | 'OLS with feature selection'                           | 59                 |
| 5     | 'Unconverged' | 'no'                | 0.965543949985539                    | 'GPR with "ardsquaredexponential" covariance function' | 378                |
| 6     | 'Converged'   | 'yes'               | 0.996501542812051                    | 'GPR with "piecewisepoly" covariance function'         | 302                |
| 7     | 'Unconverged' | 'no'                | 0.999898254264918                    | 'GPR with "ardperiodicmatern52" covariance function'   | 378                |
| 8     | 'Converged'   | 'no'                | 0.991161865273363                    | 'OLS with feature selection'                           | 67                 |

Tabela 14: Simulink Caso 2 (Lola-Voronoi)

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                  | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|--|--------------------|
| 1     | 'Converged'   | 'yes'               | 0.995156632388841                    | 'OLS with feature selection'                         | 61                 |
| 2     | 'Converged'   | 'yes'               | 0.995487218140507                    | 'OLS with feature selection'                         | 61                 |
| 3     | 'Converged'   | 'yes'               | 0.995520264535623                    | 'OLS with feature selection'                         | 61                 |
| 4     | 'Converged'   | 'yes'               | 0.996969129459752                    | 'OLS with feature selection'                         | 61                 |
| 5     | 'Unconverged' | 'no'                | 0.964560756106292                    | 'GPR with "ardgabor" covariance function'            | 388                |
| 6     | 'Converged'   | 'yes'               | 0.995780093567590                    | 'GPR with "ardperiodicmatern52" covariance function' | 298                |
| 7     | 'Unconverged' | 'no'                | 0.995715298066695                    | 'GPR with "squaredexponential" covariance function'  | 388                |
| 8     | 'Converged'   | 'yes'               | 0.991042454131312                    | 'OLS with feature selection'                         | 77                 |

Tabela 15: Simulink Caso 3 (100% linear)

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                 | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|---|--------------------|
| 1     | 'Converged'   | 'yes'               | 0.992779911717020                    | 'OLS with feature selection'                        | 59                 |
| 2     | 'Converged'   | 'yes'               | 0.992962524239706                    | 'OLS with feature selection'                        | 59                 |
| 3     | 'Converged'   | 'yes'               | 0.995300800493081                    | 'OLS with feature selection'                        | 59                 |
| 4     | 'Converged'   | 'yes'               | 0.995816221783398                    | 'OLS with feature selection'                        | 59                 |
| 5     | 'Unconverged' | 'no'                | 0.935657469337737                    | 'GPR with "squaredexponential" covariance function' | 277                |
| 6     | 'Unconverged' | 'no'                | 0.996998547731801                    | 'GPR with "squaredexponential" covariance function' | 277                |
| 7     | 'Converged'   | 'yes'               | 0.974873315720505                    | 'OLS with feature selection'                        | 277                |
| 8     | 'Converged'   | 'no'                | 0.991161865273363                    | 'OLS with feature selection'                        | 67                 |

Tabela 16: Simulink Caso 4 (100% não linear)

| Resp. | Status        | Estado Estacionário | Q <sup>2</sup> com validação cruzada | Método de regressão                                   | Número de amostras |
|-------|---------------|---------------------|--------------------------------------|---|--------------------|
| 1     | 'Converged'   | 'yes'               | 0.995676269825428                    | 'OLS with feature selection'                          | 55                 |
| 2     | 'Converged'   | 'yes'               | 0.996001037938493                    | 'OLS with feature selection'                          | 55                 |
| 3     | 'Converged'   | 'yes'               | 0.997125087808457                    | 'GPR with "squaredexponential" covariance function'   | 55                 |
| 4     | 'Converged'   | 'yes'               | 0.998989035881597                    | 'GPR with "squaredexponential" covariance function'   | 53                 |
| 5     | 'Unconverged' | 'no'                | 0.851472222626928                    | 'GPR with "ardperiodicmatern52" covariance function'  | 196                |
| 6     | 'Converged'   | 'yes'               | 0.996105396066127                    | 'GPR with "ardrationalquadratic" covariance function' | 196                |
| 7     | 'Converged'   | 'yes'               | 0.968616448159191                    | 'GPR with "squaredexponential" covariance function'   | 100                |
| 8     | 'Converged'   | 'yes'               | 0.996690432067959                    | 'GPR with "squaredexponential" covariance function'   | 59                 |

## ANEXO I

```

# ASPEN Dictionary - OUTPUTS
# Written by BRCOMM Team

# -----
# -----
## BLOCKS ##

RadFrac = {"B_PRES": ["\Data\Blocks\BlockName\Output\B_PRES", 'Pressure
Profile for each stage'],
           "B_TEMP": ["\Data\Blocks\BlockName\Output\B_TEMP",
'Temperature Profile for each stage'],
           "TOP_TEMP": ['\Data\Blocks\BlockName\Output\TOP_TEMP',
'Temperature of Condenser/Top Stage'],
           "SCTEMP": ["\Data\Blocks\BlockName\Output\SCTEMP",
'Subcooled temperature of Condenser/Top Stage'],
           "COND_DUTY": ['\Data\Blocks\BlockName\Output\COND_DUTY',
'Heat duty of Condenser/Top Stage'],
           "SCDUTY": ["\Data\Blocks\BlockName\Output\SCDUTY",
'Subcooled duty of Condenser/Top Stage'],
           "MOLE_D": ["\Data\Blocks\BlockName\Output\MOLE_D',
'Distillate rate'],
           "MOLE_L1": ["\Data\Blocks\BlockName\Output\MOLE_L1',
'Reflux rate'],
           "MOLE_RR": ["\Data\Blocks\BlockName\Output\MOLE_RR',
'Reflux ratio'],
           "MOLE_DW": ['\Data\Blocks\BlockName\Output\MOLE_DW', 'Free
water distillate rate'],
           "RW": ['\Data\Blocks\BlockName\Output\RW', 'Free water
reflux ratio'],
           "MOLE_DFR": ['\Data\Blocks\BlockName\Output\MOLE_DFR',
'Distillate to feed ratio'],
           "BOTTOM_TEMP":
['\Data\Blocks\BlockName\Output\BOTTOM_TEMP', 'Temperature of Reboiler
Bottom Stage'],
           "REB_DUTY": ['\Data\Blocks\BlockName\Output\REB_DUTY',
'Heat duty of Reboiler Bottom Stage'],
           "MOLE_B": ['\Data\Blocks\BlockName\Output\MOLE_B', 'Bottoms
rate'],
           "MOLE_VN": ["\Data\Blocks\BlockName\Output\MOLE_VN',
'Boilup rate'],
           "MOLE_BR": ["\Data\Blocks\BlockName\Output\MOLE_BR',
'Boilup ratio'],
           "MOLE_BFR": ["\Data\Blocks\BlockName\Output\MOLE_BFR',
'Bottoms to feed ratio']}]

Mixer = {"B_TEMP": ['\Data\Blocks\BlockName\Output\B_TEMP', 'Outlet
temperature'],
         "B_PRES": ['\Data\Blocks\BlockName\Output\B_PRES', 'Outlet
pressure'],
         "B_VFRAC": ['\Data\Blocks\BlockName\Output\B_VFRAC', 'Vapor
fraction'],
         "LIQ_RATIO": ['\Data\Blocks\BlockName\Output\LIQ_RATIO', '1st
liquid/total liquid'],
         "PDROP": ['\Data\Blocks\BlockName\Output\PDROP', 'Pressure
drop']}

```



```

Flash2 = {"B_TEMP": ['\Data\Blocks\BlockName\Output\B_TEMP', 'Outlet
temperature'],
          "B_PRES": ['\Data\Blocks\BlockName\Output\B_PRES', 'Outlet
pressure'],
          "B_VFRAC": ['\Data\Blocks\BlockName\Output\B_VFRAC', 'Vapor
fraction (mole)'],
          "MVFRAC": ['\Data\Blocks\BlockName\Output\MVFRAC', 'Vapor
fraction (mass)'],
          "QCALC": ['\Data\Blocks\BlockName\Output\QCALC', 'Heat
duty'],
          "QNET": ['\Data\Blocks\BlockName\Output\QNET', 'Net Duty'],
          "LIQ_RATIO": ['\Data\Blocks\BlockName\Output\LIQ_RATIO', '1st
liquid/total liquid'],
          "PDROP": ['\Data\Blocks\BlockName\Output\PDROP', 'Pressure
drop']}

Heater = {"B_TEMP": ['\Data\Blocks\BlockName\Output\B_TEMP', 'Outlet
temperature'],
          "B_PRES": ['\Data\Blocks\BlockName\Output\B_PRES', 'Outlet
pressure'],
          "B_VFRAC": ['\Data\Blocks\BlockName\Output\B_VFRAC', 'Vapor
fraction'],
          "QCALC": ['\Data\Blocks\BlockName\Output\QCALC', 'Heat
duty'],
          "QNET": ['\Data\Blocks\BlockName\Output\QNET', 'Net Duty'],
          "LIQ_RATIO": ['\Data\Blocks\BlockName\Output\LIQ_RATIO', '1st
liquid/total liquid'],
          "PDROP": ['\Data\Blocks\BlockName\Output\PDROP', 'Pressure
drop']}

FSplit                                     =                               {"STREAMFRAC":
['\Data\Blocks\BlockName\Output\STREAMFRAC\StreamName', 'Split
fraction'],
          "STREAM_ORDER":
['\Data\Blocks\BlockName\Output\STREAM_ORDER\StreamName', 'Stream
Order']}

Pump = {"FLUID_POWER": ['\Data\Blocks\BlockName\Output\FLUID_POWER',
'Fluid Power'],
        "BRAKE_POWER": ['\Data\Blocks\BlockName\Output\BRAKE_POWER',
'Brake Power'],
        "ELEC_POWER": ['\Data\Blocks\BlockName\Output\ELEC_POWER',
'Electricity'],
        "VFLOW": ['\Data\Blocks\BlockName\Output\VFLOW', 'Volumetric
Flow Rate'],
        "PDRP": ['\Data\Blocks\BlockName\Output\PDRP', 'Pressure
Change'],
        "NPSH-AVAIL": [r'\Data\Blocks\BlockName\Output\NPSH-AVAIL',
'NPSH Available'],
        "HEAD_CAL": ['\Data\Blocks\BlockName\Output\HEAD_CAL', 'Head
Developed'],
        "CEFF": ['\Data\Blocks\BlockName\Output\CEFF', 'Pump Efficiency
used'],
        "WNET": ['\Data\Blocks\BlockName\Output\WNET', 'Net Work
Required'],
        "POC": ['\Data\Blocks\BlockName\Output\POC', 'Outlet
Pressure'],
        "TOC": ['\Data\Blocks\BlockName\Output\TOC', 'Outlet
Temperature']}

```

```

Compr = {"IND_POWER": ['\Data\Blocks\BlockName\Output\IND_POWER',
'Indicated horsepower'],
        "BRAKE_POWER": ['\Data\Blocks\BlockName\Output\BRAKE_POWER',
'Brake horsepower'],
        "WNET": ['\Data\Blocks\BlockName\Output\WNET', 'Net work
required'],
        "POWER_LOSS": ['\Data\Blocks\BlockName\Output\POWER_LOSS',
'Power loss'],
        "EPC": ['\Data\Blocks\BlockName\Output\EPC', 'Efficiency'],
        "EFF_MECH": ['\Data\Blocks\BlockName\Output\EFF_MECH',
'Mechanical efficiency'],
        "POC": ['\Data\Blocks\BlockName\Output\POC', 'Outlet
pressure'],
        "TOC": ['\Data\Blocks\BlockName\Output\TOC', 'Outlet
temperature'],
        "TOS": ['\Data\Blocks\BlockName\Output\TOS', 'Isentropic
outlet temperature'],
        "B_VFRAC": ['\Data\Blocks\BlockName\Output\B_VFRAC', 'Vapor
fraction'],
        "DIS": ['\Data\Blocks\BlockName\Output\DIS', 'Displacement'],
        "EV": ['\Data\Blocks\BlockName\Output\EV', 'Volumetric
efficiency']}

```

```

HeatX = {"HOTINT": ['\Data\Blocks\BlockName\Output\HOTINT',
'Temperature of the inlet hot stream'],
        "HOTINP": ['\Data\Blocks\BlockName\Output\HOTINP', 'Pressure
of the inlet hot stream'],
        "HOTINVF": ['\Data\Blocks\BlockName\Output\HOTINVF', 'Vapor
fraction of the inlet hot stream'],
        "HIN_L1FRAC": ['\Data\Blocks\BlockName\Output\HIN_L1FRAC',
'1st liquid/total liquid of the inlet hot stream'],
        "COLDINT": ['\Data\Blocks\BlockName\Output\COLDINT',
'Temperature of the inlet cold stream'],
        "COLDINP": ['\Data\Blocks\BlockName\Output\COLDINP', 'Pressure
of the inlet cold stream'],
        "COLDINVF": ['\Data\Blocks\BlockName\Output\COLDINVF', 'Vapor
fraction of the inlet cold stream'],
        "CIN_L1FRAC": ['\Data\Blocks\BlockName\Output\CIN_L1FRAC',
'1st liquid/total liquid of the inlet cold stream'],
        "HOT_TEMP": ['\Data\Blocks\BlockName\Output\HOT_TEMP',
'Temperature of the outlet hot stream'],
        "HOT_PRES": ['\Data\Blocks\BlockName\Output\HOT_PRES',
'Pressure of the outlet hot stream'],
        "HOT_VFRAC": ['\Data\Blocks\BlockName\Output\HOT_VFRAC',
'Vapor fraction of the outlet hot stream'],
        "HOUT_L1FRAC": ['\Data\Blocks\BlockName\Output\HOUT_L1FRAC',
'1st liquid/total liquid of the outlet hot stream'],
        "COLD_TEMP": ['\Data\Blocks\BlockName\Output\COLD_TEMP',
'Temperature of the outlet cold stream'],
        "COLD_PRES": ['\Data\Blocks\BlockName\Output\COLD_PRES',
'Pressure of the outlet cold stream'],
        "COLD_FRAC": ['\Data\Blocks\BlockName\Output\COLD_FRAC',
'Vapor fraction of the outlet cold stream'],
        "COUT_L1FRAC": ['\Data\Blocks\BlockName\Output\COUT_L1FRAC',
'1st liquid/total liquid of the outlet cold stream']}

```

```

Valve = {"P_OUT_OUT": ['\Data\Blocks\BlockName\Output\P_OUT_OUT',
'Outlet Pressure'],
        "VALVE_DP": ['\Data\Blocks\BlockName\Output\VALVE_DP',
'Pressure Drop'],

```

```

        "TCALC": ['\Data\Blocks\BlockName\Output\TCALC', 'Outlet
Temperature'],
        "VCALC": ['\Data\Blocks\BlockName\Output\VCALC', 'Outlet Vapor
Fraction'],
        "PIPE_FIT_FAC2":
['\Data\Blocks\BlockName\Output\PIPE_FIT_FAC2', 'Piping Geometry
Factor']]

Flash3 = {"B_TEMP": ['\Data\Blocks\BlockName\Output\B_TEMP', 'Outlet
Temperature'],
        "B_PRES": ['\Data\Blocks\BlockName\Output\B_PRES', 'Outlet
Pressure'],
        "B_VFRAC": ['\Data\Blocks\BlockName\Output\B_VFRAC', 'Vapor
Fraction (mole)'],
        "MVFRAC": ['\Data\Blocks\BlockName\Output\MVFRAC', 'Vapor
Fraction (mass)'],
        "QCALC": ['\Data\Blocks\BlockName\Output\QCALC', 'Heat
Duty'],
        "QNET": ['\Data\Blocks\BlockName\Output\QNET', 'Net Duty'],
        "LIQ_RATIO": ['\Data\Blocks\BlockName\Output\LIQ_RATIO', '1st
Liquid/Total Liquid'],
        "PDROP": ['\Data\Blocks\BlockName\Output\PDROP', 'Pressure
Drop']]

RPlug = {"QCALC": ['\Data\Blocks\BlockName\Output\QCALC', 'Reactor
Calculated Heat Duty.'],
        "TMIN": ['\Data\Blocks\BlockName\Output\TMIN', 'Reactor Lowest
Temperature.'],
        "TMAX": ['\Data\Blocks\BlockName\Output\TMIN', 'Reactor
Highest Temperature.'],
        "RES_TIME": ['\Data\Blocks\BlockName\Output\RES_TIME',
'Reactor Residence Time.']}

RCSTR = {"QCALC": ['\Data\Blocks\BlockName\Output\QCALC', 'Reactor
Calculated Heat Duty.'],
        "TOT_VOL": ['\Data\Blocks\BlockName\Output\TOT_VOL', 'Reactor
Total Volume.'],
        "TOT_RES_TIME": ['\Data\Blocks\BlockName\Output\TOT_RES_TIME',
'Reactor Total Res. Time.'],
        "VAP_VOL": ['\Data\Blocks\BlockName\Output\VAP_VOL', 'Reactor
Vapor Phase Volume.'],
        "LIQ_VOL": ['\Data\Blocks\BlockName\Output\LIQ_VOL', 'Reactor
Liquid Phase Volume.']}
# -----
# STREAMS ##

MATERIAL = {"TEMP_OUT":
['\Data\Streams\StreamName\Output\TEMP_OUT\MIXED', 'Temperature'],
        "PRES_OUT":
['\Data\Streams\StreamName\Output\PRES_OUT\MIXED', 'Pressure'],
        "VFRAC_OUT":
['\Data\Streams\StreamName\Output\VFRAC_OUT\MIXED', 'Molar Vapor
Fraction'],
        "LFRAC": ['\Data\Streams\StreamName\Output\LFRAC\MIXED',
'Molar Liquid Fraction'],
        "SFRAC": ['\Data\Streams\StreamName\Output\SFRAC\MIXED',
'Molar Solid Fraction'],
        "MASSVFRA": ['\Data\Streams\StreamName\Output\MASSVFRA',
'Mass Vapor Fraction'],

```

```

        "MASSSFRA":      ['\Data\Streams\StreamName\Output\MASSSFRA',
'Mass Solid Fraction'],
        "HMX": ['\Data\Streams\StreamName\Output\HMX\MIXED', 'Molar
Enthalpy'],
        "HMX_MASS":
['\Data\Streams\StreamName\Output\HMX_MASS\MIXED', 'Mass Enthalpy'],
        "SMX": ['\Data\Streams\StreamName\Output\SMX\MIXED', 'Molar
Entropy'],
        "SMX_MASS":
['\Data\Streams\StreamName\Output\SMX_MASS\MIXED', 'Mass Entropy'],
        "RHOMX":      ['\Data\Streams\StreamName\Output\RHOMX\MIXED',
'Molar Density'],
        "RHOMX_MASS":
['\Data\Streams\StreamName\Output\RHOMX_MASS\MIXED', 'Mass Density'],
        "HMX_FLOW":
['\Data\Streams\StreamName\Output\HMX_FLOW\MIXED', 'Enthalpy Flow'],
        "MWMX":      ['\Data\Streams\StreamName\Output\MWMX\MIXED',
'Average Molecular Weight'],
        "MOLEFLMX":
['\Data\Streams\StreamName\Output\MOLEFLMX\MIXED', 'Total Mole Flow'],
        "MOLEFLOW":
['\Data\Streams\StreamName\Output\MOLEFLOW\MIXED', 'Mole Flow of
Component'],
        "MOLEFRAC":
['\Data\Streams\StreamName\Output\MOLEFRAC\MIXED', 'Mole Fraction of
Component'],
        "MASSFLMX":
['\Data\Streams\StreamName\Output\MASSFLMX\MIXED', 'Total Mass Flow'],
        "MASSFLOW":
['\Data\Streams\StreamName\Output\MASSFLOW\MIXED', 'Mass Flow of
Component'],
        "MASSFRAC":
['\Data\Streams\StreamName\Output\MASSFRAC\MIXED', 'Mass Fraction of
Component'],
        "VOLFLMX":
['\Data\Streams\StreamName\Output\VOLFLMX\MIXED', 'Total Volume Flow']}

```