



Universidade Federal da Bahia  
Escola Politécnica / Instituto de Computação

Programa de Pós-Graduação em Mecatrônica

**MODELING AND UNCERTAINTY  
ASSESSMENT OF DYNAMICAL SYSTEMS  
AND DIGITAL TWINS**

Erbet Almeida Costa

TESE DE DOUTORADO

Salvador  
13 de junho de 2024



ERBET ALMEIDA COSTA

**MODELING AND UNCERTAINTY ASSESSMENT OF  
DYNAMICAL SYSTEMS AND DIGITAL TWINS**

Esta Tese de Doutorado foi apresentada ao Programa de Pós-Graduação em Mecatrônica da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutor em Mecatrônica.

Orientador: Prof. Dr. Leizer Schnitman  
Co-orientador: Prof. Dr. Idelfonso B. R. Nogueira

Salvador  
13 de junho de 2024

Ficha catalográfica elaborada pela Biblioteca Bernadete  
Sinay Neves, Escola Politécnica - UFBA.

---

C837 Costa, Erbet Almeida.

Modeling and uncertainty assessment of dynamical systems  
and digital twins/ Erbet Almeida Costa. – Salvador, 2024.

163f.: il. color.

Orientador: Prof. Dr. Leizer Schnitman.

Coorientador: Prof. Dr. Idelfonso Bessa dos Reis Nogueira.

Tese (doutorado) – Programa de Pós-Graduação em  
Mecatrônica, Escola Politécnica e Instituto de Computação,  
Universidade Federal da Bahia, 2024.

1. Inteligência artificial. 2. Modelagem. 3. Simulação. 4.  
Incerteza. 5. Gêmeo digital. I. Schnitman, Leizer. II. Nogueira,  
Idelfonso Bessa dos Reis. III. Universidade Federal da Bahia. IV.  
Título.

---

CDD: 629.8



ERBET ALMEIDA COSTA

**MODELING AND UNCERTAINTY ASSESSMENT OF DYNAMICAL SYSTEMS  
AND DIGITAL TWINS**

Tese submetida ao corpo docente do programa de pós-graduação em Mecatrônica da Universidade Federal da Bahia como parte dos requisitos necessários para a obtenção do grau de doutor em Mecatrônica.

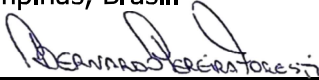
Salvador, 13 de junho de 2024.


Banca Examinadora

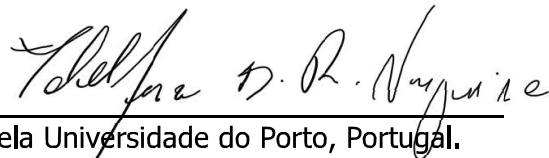
Prof. Dr. Francesco Corona   
Doutor em Engenharia Industrial, Università degli Studi di Cagliari, Itália.  
Universidade Aalto, Finlândia.

Prof. Dr. Erick Giovanni Sperandio Nascimento   
Doutorado em Engenharia Ambiental pela Universidade Federal do Espírito Santo, Brasil.  
University of Surrey, Reino Unido.

Prof. Dr. Flávio Vasconcelos da Silva   
Doutor em Engenharia de Alimentos pela Universidade Estadual de Campinas, Brasil.  
Universidade Estadual de Campinas, Brasil.

Dr. Bernardo Pereira Foresti   
Doutor em Engenharia Mecânica pela Universidade Estadual de Campinas, Brasil.  
Petrobras Cenpes, Brasil.

Prof. Dr. Leizer Schnitman   
Doutorado em Engenharia Eletrônica e Computação pelo Instituto Tecnológico de Aeronáutica, Brasil.  
Universidade Federal da Bahia, Brasil.

Prof. Dr. Idelfonso Bessa dos Reis Nogueira   
Doutor em Engenharia Química e Biológica pela Universidade do Porto, Portugal.  
Universidade Norueguesa de Ciência e Tecnologia, Noruega.



## ACKNOWLEDGMENTS

With all my heart, I express my deepest gratitude to my family: my wife, Carine Rebello; my parents, Valdenice and Sebastião; and my siblings, Janair, Eder, and Jesana. Your unwavering love, support, and belief in me have been the bedrock of my journey and the source of my strength and motivation.

I am immensely grateful to my supervisors, Dr. Leizer Schnitman and Idelfonso Nogueira, for their invaluable guidance and insights throughout this research journey. Your mentorship has shaped my work and my development as a scientist.

I am also thankful to my colleagues from CTAI, namely, Tiago, Odilon, Galdir, Lúcia, Fontana, Flávio and Prof. Ana Mafalda from FEUP, and others who supported me in this journey for the enriching discussions and collaborative spirit that greatly enhanced my research experience.

My research would not have been possible without the generous financial support from Capes, Petrobras, LSRE-LCM, and ALiCE laboratories at Porto University, whom I thank very much.

I extend my appreciation to all the participants and institutions involved in my research, whose contributions were crucial to its success.

I am grateful to my friends, both within and outside the academic realm, for their continuous encouragement and for providing moments of joy and respite during the challenging times of this journey.

Lastly, I thank my psychologist, who helped me learn how to handle some personal issues and contributed (a lot) to completing this thesis.

This Ph.D. journey has been as challenging as it has been rewarding, and I am grateful to everyone who has participated.



*Ninguém sai da vida tal qual entrou.*

—SÊNECA (Aprendendo a viver)



## RESUMO

Esta tese apresenta uma metodologia inovadora destinada a atender às crescentes demandas e desafios em sistemas de Aprendizado de Máquina Científica (SciML, do inglês *Scientific Machine Learning*), sistemas de bombeamento centrífugo submerso (ESP, do inglês *Electric Submersible Pumps*), sistemas de elevação de artificial e processos de adsorção por oscilação de pressão (PSA) por meio de aprendizado robusto, estimação de parâmetros, aprendizagem profunda e gêmeos digitais. A metodologia proposta é abrangente para avaliar múltiplas facetas de incerteza inerentes à identificação do modelo SciML, considerando a base da literatura, a sensibilidade dos dados e o esforço computacional. A metodologia identifica e valida modelos de aprendizagem profunda, utilizando um modelo não lineares para geração de dados e superação de limitações de dados experimentais. A metodologia possui um método Bayesiano integrado, como etapa metodológica, que é empregado para estimação de parâmetros, avaliação da incerteza, validação de modelos fenomenológicos e orientado a dados. O método é tratado em etapas que alinham com sucesso o modelo com dados experimentais, tanto dinamicamente como em estado estacionário, mostrando o potencial da metodologia para representar o comportamento do sistema dentro das incertezas existentes. Este desenvolvimento permite a construção de modelos dinâmicos de IA confiáveis e computacionalmente eficientes para planejamento, construção, controle e otimização de gêmeos digitais. A metodologia é posta a prova frente diversos estudos de caso. Os resultados são compostos por modelos validados contra dados sintéticos e experimentais que apresentam compatibilidade com o comportamento dinâmico do modelo não linear e suas incertezas. Uma primeira validação do método é realizada através de um estudo de caso envolvendo o desenvolvimento de um sensor virtual para um reator de polimerização, demonstrando robustez e consistência no tratamento de incertezas no campo SciML. Dois estudos de caso são realizados para sistemas de elevação artificial baseado em BCS. Nestes estudos de casos a técnica se mostrou promissora para caracterização e representação do sistema e pavimentou o caminho para aplicações em campos de produção de petróleo, particularmente em controle de produção, otimização e assistência. No contexto dos processos PSA, é apresentada uma nova abordagem para o desenvolvimento de um gêmeo digital com avaliação da incerteza, capaz de mapear o comportamento cíclico e complexo dos sistemas PSA. Através da aprendizagem on-line contínua e da integração de um novo *feedback tracker*, o gêmeo digital representa e se adapta com precisão às complexidades do sistema PSA, abordando desafios como a degradação do adsorvente. Esta metodologia oferece repostas sobre as aplicações da IA e dos gêmeos digitais na otimização de processos industriais e no apoio ao desenvolvimento sustentável em vários setores. Juntos, esses trabalhos contribuem com resultados e metodologias valiosas para seus respectivos campos, demonstrando o potencial das tecnologias avançadas para melhorar a representação de sistemas, abordar

incertezas e abrir caminho para desenvolvimentos futuros em aplicações industriais.

**Palavras-chave:** Modelagem, Simulação, Incerteza, Inteligência Artificial, Gêmeo Digital.



## ABSTRACT

This thesis presents an innovative methodology aimed to meet the growing demands and challenges in Scientific Machine Learning (SciML) applied to electrical submersible pumps (ESP) and pressure swing adsorption (PSA) processes through robust learning, parameter estimation, deep learning, and digital twins. The proposed methodology comprehensively evaluates multiple facets of uncertainty inherent to identifying the SciML model, considering the literature base, data sensitivity, and computational effort. The methodology identifies and validates deep learning models, using non-linear models to generate and overcome experimental data limitations. The methodology uses an integrated Bayesian method as a methodological step to estimate parameters, assess uncertainty, and validate phenomenological and data-driven models. The method is treated in steps that successfully align the model with experimental data, both dynamically and in a steady state, showing the methodology's potential to represent the system's behavior within existing uncertainties. This development enables the construction of reliable and computationally efficient dynamic AI models for planning, building, controlling, and optimizing digital twins. The methodology is put under test in several case studies. The results are composed of models validated against synthetic and experimental data that are compatible with the dynamic behavior of the nonlinear model and its uncertainties. The first validation of the method is carried out through a case study involving the development of a soft sensor for a polymerization reactor, which demonstrates robustness and consistency in the treatment of uncertainties in the SciML field. Two case studies are performed for ESP-based artificial lift systems. In those case studies, the technique showed promise for the characterization and representation of the system and paves the way for applications in oil production fields, particularly in production control, optimization, and assistance. In the context of PSA processes, a new approach is presented for developing a digital twin with uncertainty assessment capable of mapping PSA systems' cyclical and complex behavior. Through continuous online learning and integrating a new feedback tracker, the digital twin accurately represents and adapts to the complexities of the PSA system, addressing challenges such as adsorbent degradation. This methodology offers answers about the applications of AI and digital twins in optimizing industrial processes and supporting sustainable development in various sectors. Together, these works contribute valuable results and methodologies to their respective fields, demonstrating the potential of advanced technologies to improve system representation, address uncertainties, and pave the way for future developments in industrial applications.

**Keywords:** Modeling, Simulation, Markov Chain Monte Carlo, Artificial Intelligence, Digital Twin.



# CONTENTS

<b>Chapter 1—Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Objectives . . . . .	2
1.3 Thesis structure . . . . .	2
1.4 Literature review . . . . .	3
1.4.1 Electric submersible pumps . . . . .	4
1.4.2 Pressure swing absorption . . . . .	8
1.5 Basic concepts . . . . .	10
1.5.1 Parameter estimation and model uncertainty assessment . . . . .	10
1.5.1.1 Bayesian inference . . . . .	12
1.5.2 Neural networks for system identification . . . . .	14
1.5.2.1 Definitions . . . . .	16
1.5.2.2 Loss functions . . . . .	19
1.5.2.3 Solving algorithms . . . . .	20
1.5.3 Uncertainty assessment Experimental data . . . . .	20
1.5.3.1 Auto correlation . . . . .	21
1.5.3.2 Type A uncertainty evaluation . . . . .	21
1.5.3.3 Type B uncertainty evaluation . . . . .	22
1.5.3.4 Dependent variables . . . . .	23
1.5.3.5 Uncertainty assessment results . . . . .	23
1.5.3.6 Coverage factor . . . . .	23
<b>Chapter 2—A robust learning methodology for uncertainty-aware scientific machine learning models</b>	<b>25</b>
2.1 Introduction . . . . .	25
2.2 Methodology for the Monte Carlo uncertainty training . . . . .	25
2.2.1 Markov chain Monte Carlo . . . . .	26
2.2.2 Synthetic data generation . . . . .	29
2.2.3 Data curation . . . . .	30
2.2.4 Building the AI model . . . . .	31
2.2.5 Monte Carlo training . . . . .	31
2.2.6 Propagation and cross-validation . . . . .	32
2.3 Results and discussion . . . . .	34
2.3.1 Case study: Polymerization reactor . . . . .	35
2.3.2 MCMC . . . . .	39

2.3.3	Synthetic data generation for training . . . . .	39
2.3.4	Monte Carlo training . . . . .	40
2.3.4.1	Monte Carlo training . . . . .	40
2.3.4.2	Data size . . . . .	41
2.3.4.3	Training . . . . .	42
2.3.5	Uncertainty propagation and validation . . . . .	44
2.4	Conclusion . . . . .	45

### **Chapter 3—A Bayesian approach to the dynamic modeling of ESP-lifted oil well systems: An experimental validation on an ESP prototype.** 47

3.1	Introduction . . . . .	47
3.2	Methodology . . . . .	47
3.2.1	Bayesian method for parameter estimation . . . . .	49
3.2.2	Evaluation of the uncertainty of the experimental data . . . . .	50
3.2.2.1	Steady-state . . . . .	51
3.2.2.2	Autocorrelation . . . . .	51
3.3	Results and discussions . . . . .	51
3.3.1	Experimental apparatus . . . . .	51
3.3.2	Modeling . . . . .	53
3.3.3	Estimation problem and model adaptation . . . . .	55
3.3.4	Evaluation of uncertainty of experimental data . . . . .	57
3.3.5	Parameter estimation and validation . . . . .	58
3.3.6	Validation . . . . .	61
3.3.7	Test . . . . .	62
3.4	Conclusion . . . . .	64

### **Chapter 4—An uncertainty approach for Electric Submersible Pump modeling through Deep Neural Network** 67

4.1	Introduction . . . . .	67
4.2	Methodology . . . . .	67
4.3	Case study . . . . .	68
4.3.1	Data augmentation . . . . .	68
4.3.2	Data curation . . . . .	70
4.3.3	DNN hyperparameters and training . . . . .	72
4.3.4	MCMC training . . . . .	73
4.3.5	Propagation and experimental validation . . . . .	80
4.4	Conclusions . . . . .	81

### **Chapter 5—Adaptive digital twin for pressure swing adsorption systems\*** 85

5.1	Introduction . . . . .	85
5.2	Methodology and results . . . . .	85
5.2.1	Pressure swing adsorption virtual plant . . . . .	86

CONTENTS	xv
5.2.2 Digital twining - Development steps . . . . .	91
5.2.3 Digital twining - Application tests . . . . .	96
5.3 Conclusions . . . . .	101
<b>Chapter 6—Conclusions</b>	<b>103</b>
6.1 Final remarks . . . . .	103
6.2 Future work . . . . .	104
<b>Appendix</b>	<b>114</b>
<b>Appendix A—Supplementary analysis of MCMC chains</b>	<b>115</b>



## LIST OF FIGURES

1.1	Perceptron scheme. . . . .	15
1.2	Multilayer perceptron. . . . .	15
1.3	Long-Short-Term Memory cell. . . . .	18
2.1	General methodology scheme. . . . .	26
2.2	MCMC method. . . . .	28
2.3	Monte Carlo simulation method to data generating. . . . .	30
2.4	Optimization procedure to find the hyperparameters of the neural network. . . . .	32
2.5	Monte Carlo training method. . . . .	33
2.6	LHS inputs. . . . .	37
2.7	Inputs correlation map. . . . .	38
2.8	Synthetic output data with white noise. . . . .	38
2.9	a) Lipschitz surface for reactor temperature; b) Lipschitz surface for polymer viscosity. . . . .	40
2.10	Training performance as a function of experiments. (—) Validation data; (—) Training data. . . . .	42
2.11	Training performance as a function of experiments. . . . .	43
2.12	Trained epochs of Monte Carlo Training. . . . .	44
2.13	Training and validation data of $T$ . (—) Nonlinear model; (—) NN model, (—) Nonlinear model uncertainty, (—) NN model uncertainty. . . . .	45
2.14	Training and validation data of $\eta$ . (—) Nonlinear model; (—) NN model, (—) Nonlinear model uncertainty, (—) NN model uncertainty. . . . .	46
3.1	Proposed method for the validation of ESP models. . . . .	48
3.2	Autocorrelation of production flow with a sampling time of 1 second. . . . .	52
3.3	Autocorrelation of flow with a sampling time of 100 seconds. . . . .	52
3.4	ESP-lifted well set up. . . . .	53
3.5	Inputs signal inserted in the prototype. . . . .	59
3.6	Sensitivity analysis of the likelihood function. . . . .	60
3.7	Confidence bound of estimated parameters. . . . .	60
3.8	Intake level - direct validation. . . . .	62
3.9	Wellhead pressure - direct validation. . . . .	63
3.10	Production flow - direct validation. . . . .	63
3.11	Inputs signal inserted in the prototype to test dataset. . . . .	64
3.12	Production flow in the test dataset. . . . .	65
3.13	Wellhead pressure in the test dataset. . . . .	65
3.14	Production flow in test dataset. . . . .	66

4.1	Schematic diagram of the proposed method. Dotted lines are models or parameters, and solid lines are data. . . . .	68
4.2	Uncertainty propagation on the nonlinear model. . . . .	69
4.3	PRBS signal inputted in the nonlinear model. . . . .	70
4.4	Lipschitz Quotients for identification of the Inputs and Outputs order. . .	71
4.5	Choke pressure DNN MAE and MSE training and test values. . . . .	74
4.6	Intake pressure DNN MAE and MSE training and test values. . . . .	75
4.7	Production flow DNN MAE and MSE training and test values. . . . .	76
4.8	MAE and MSE experiment varying the amount of data for choke valve pressure. . . . .	77
4.9	MAE and MSE experiment varying the amount of data for intake pressure. .	77
4.10	MAE and MSE experiment varying the amount of data for production flow. .	78
4.11	Monte Carlo training method. . . . .	78
4.12	Parity chart between the measured and DNN predicted values. In (A) is the intake pressure, in (B) is the choke pressure, and in (C) is the flow rate. .	79
4.13	Chain analysis of $P_c$ . . . . .	80
4.14	Chain analysis of $P_{in}$ . . . . .	81
4.15	Chain analysis of $Q_c$ . . . . .	82
4.16	Input variables: (A) ESP motor frequency ( $f$ ), (B) valve opening level ( $Z_c$ ), (C) reservoir pressure ( $P_r$ ), and manifold pressure ( $P_m$ ). . . . .	83
4.17	Experimental validation of $P_c$ . . . . .	83
4.18	Experimental validation of $P_{in}$ . . . . .	84
4.19	Experimental validation of $Q_c$ . . . . .	84
5.1	General methodology flowchart. The abbreviations are: LHS - Latin Hypercube Sampling, DNN - Deep neural network, MCMC - Markov Chain Monte Carlo, PDF - Probability Density Function, FB - Feedback Tracker, OL - Online Learning. . . . .	87
5.2	Pressure swing adsorption stages. . . . .	88
5.3	Correlation map of inputs. . . . .	93
5.4	Lipschitz index values map. . . . .	94
5.5	$H_2/CO$ DNN parameters MCMC analysis. . . . .	97
5.6	$CO_2$ recovery DNN parameters MCMC analysis. . . . .	97
5.7	$CO_2$ purity DNN parameters MCMC analysis. . . . .	98
5.8	Digital Twin application to $CO_2$ purity. . . . .	100
5.9	Digital Twin application to $CO_2$ recovery. . . . .	101
5.10	Digital Twin application to $H_2$ and $CO$ ratio. . . . .	102
A.1	Coverage regions of parameters 01. . . . .	115
A.2	Coverage regions of parameters 02. . . . .	115
A.3	Coverage regions of parameters 03. . . . .	116
A.4	Coverage regions of parameters 04. . . . .	116
A.5	Coverage regions of parameters 05. . . . .	116
A.6	Coverage regions of parameters 06. . . . .	117



A.7 Coverage regions of parameters 07. . . . . 117

A.8 Coverage regions of parameters 08. . . . . 117

A.9 Coverage regions of parameters 09. . . . . 118

A.10 Coverage regions of parameters 10. . . . . 118

A.11 Parameters random walk. . . . . 118



## LIST OF TABLES

1.1	Examples of activation functions. . . . .	16
2.1	Parameters and initial conditions. . . . .	36
2.2	Steady-state inputs conditions and LHS region. . . . .	36
2.3	Output variables at steady-state. . . . .	37
2.4	Normalized parameters obtained by DRAM algorithm. . . . .	40
2.5	Hyperband hyperspace search. . . . .	41
2.6	Resulting networks hyperparameters. . . . .	41
2.7	Resulting networks hyperparameters. . . . .	44
3.1	Model variables . . . . .	56
3.2	Model parameters . . . . .	57
3.3	Uncertainty evaluation results . . . . .	58
3.4	Initial guess estimation for the parameters. . . . .	59
3.5	MCMC Statistics . . . . .	61
4.1	Setup of the Hyperband algorithm. . . . .	72
4.2	Results found by the Hyperband algorithm. . . . .	73
4.3	Geweke statistics. . . . .	79
5.1	Inputs variables limits. . . . .	88
5.2	Langmuir fitting parameters of the model for CO <sub>2</sub> , CO and H <sub>2</sub> adsorption equilibrium on MIL-125(Ti) <sub>NH2</sub> . . . . .	90
5.3	Boundary conditions. . . . .	91
5.4	Physical parameters and values . . . . .	92
5.5	Hyperband search space. . . . .	95



## CHAPTER 1

# INTRODUCTION

The recent strides in the development and training of artificial intelligence models have brought about a significant transformation in the landscape of process engineering. These advancements have not only improved the accuracy and efficiency of models but have also made previously unfeasible applications viable, thanks to the exponential growth in computational power and the subsequent decrease in training costs.

The integration of artificial intelligence in the process industry holds substantial potential for enhancement. These algorithms can be utilized in various applications, ranging from model construction to implementing controllers that require less computational effort and can resolve issues in real time. Moreover, the learning capabilities of these models enable the development of process monitoring tools that offer more sophisticated analyses. Such analyses can integrate insights based on expert knowledge or economic factors, which significantly influence the performance of the studied industrial processes.

Moreover, the uncertainty and predictive capability in models are crucial. The uncertainty inherent in a model can directly affect process performance, as models often guide controllers or system optimization. Thus, when a model is used to optimize an industrial system, any prediction uncertainty is propagated throughout the optimization process. Understanding the implications of this uncertainty propagation is vital to ensure the process is accurately directed toward the desired outcomes.

### 1.1 MOTIVATION

The primary motivation of this thesis is to assess the feasibility of constructing and applying artificial intelligence models for dynamic system modeling with uncertainty assessment. The central hypotheses tested include:

1. Uncertainty in models is assessed using established methods from the literature, applicable to both phenomenological and data-driven models.
2. The development of artificial intelligence models for processes is be enhanced using synthetic data obtained from rigorously validated models, as process data often lacks extensive operational range and variability.
3. Evaluating model uncertainty is beneficial not only theoretically but also practically.
4. Model uncertainty provides valuable insights for real-time applications like developing digital twins.

The evaluation of these hypotheses is detailed in the thesis chapters. Chapter 2 directly tests hypotheses 1 and 2 using a polymerization reaction model from existing

literature. The findings from these tests have practical implications for developing and training artificial intelligence models in process engineering. Chapters 3 and 4 explore these hypotheses using real data from the artificial lift process with centrifugal pumps, providing insights that can be applied in real-world scenarios. In Chapter 5, all four hypotheses are comprehensively tested when developing a digital twin for an adsorption gas separation system, demonstrating the practical application of the research.

## 1.2 OBJECTIVES

To test the previous hypotheses, this thesis aims to contribute to the literature on model building and uncertainty analysis by proposing a novel joint methodology for constructing and validating phenomenological and data-based dynamic models. The methods presented, while having a well-founded basis in the literature, introduce new perspectives and approaches. This work used them to contribute to discussions on model construction and validation, including as a central element validation with experimental data and assessment of the uncertainty of both data and model parameters. The proposed methods are tested in case studies that explore specific steps of the approaches. However, in a global strategy, they show the validation of the methodology in engineering applications. More specifically, the objectives can be divided into:

1. Propose a complete methodology for building artificial intelligence models and digital twins considering different sources of uncertainty;
2. Present methodology for building artificial intelligence models through the use of synthetic data provided by experimentally validated models;
3. Contribute to the electrical submersible pump systems modeling literature, providing two modeling approaches, the first being phenomenological and the second being data-driven. In both cases, the proposed models are validated with experimental data;
4. Propose a methodology for building robust digital twins of Pressure Swing Adsorption systems considering prediction uncertainty. In the proposed method, uncertainty is a key factor in activating the update steps of digital twin prediction models.

## 1.3 THESIS STRUCTURE

This thesis is structured as follows. It begins with Chapter 2, which presents the general methodology for building uncertainty-aware scientific machine-learning models and provides the primary methods for the other chapters. Chapter 3 introduces the first application of the process to develop and validate the phenomenological model of an ESP system. Chapter 4 covers applying the method to build a deep neural network model for the ESP system. Chapter 5 presents a different methodology approach to building an adaptive DT for a pressure swing adsorption system. Finally, chapter 6 concludes the thesis with general conclusions and suggestions for future work.

xchapterLiterature review and basics concepts

## 1.4 LITERATURE REVIEW

Dynamic models of physical systems have multiple uses in engineering. In many applications, models are used as surrogates in tests that can affect the real systems and are used to allow control, optimization, planning, and other activities that could have high computational effort. Those dynamic models can be built following different approaches. The most common models in process engineering are based on mass, energy, and momentum balance, providing systems of linear or nonlinear ordinary differential equations (ODEs), transfer functions, and state space. On the other hand, tools for building data-oriented models have been highlighted for modeling complex process systems.

The model's validation is essential to process systems modeling, whether phenomenological or data-driven. The most common validation method is comparing predicted values by the model with experimental values. This approach allows one to understand the model's representability given the reality of the experimental data. In addition, the model's uncertainty needs to be assessed and used as a guarantee that the model can be considered valid.

The model uncertainty can be assessed through several methods and imposing a series of assumptions that need to be made explicit. Among the available methods are those of maximum likelihood proposed by Bard (1974), Schwaab e Pinto (2007) and techniques based on Bayesian inference Gamerman e Lopes (2006), Gelman (2015). Both approaches allow for the covariance matrix of the parameters of a given model, whether dynamic or static. For some complex and rigid models, the likelihood methods may fail, as the calculation of the gradient of the likelihood function concerning the parameters may be unstable. Under these conditions, the Bayesian inference approach to assess uncertainty is a viable method to solve these problems since the solution is obtained through resampling, which avoids the calculation of derivatives. However, Bayesian methods can become computationally costly for models with many parameters, or the numerical solution of the model can be infeasible. Thus, both approaches for evaluating the uncertainty of models have pros and cons. Still, the complete validation of a model requires that the uncertainty be assessed and compared with the experimental results.

Looking at model validation, phenomenological models make up the basis for modeling process industry systems. These models are used to process improvements, control and optimization approaches, and other applications. These models' construction and validation are well established in the literature and comprise mass, energy, and momentum balances, experimental validation, and uncertainty evaluation (BARD, 1974; KLEIJNEN, 1995; SCHWAAB; PINTO, 2007).

On the other hand, machine learning (ML) has been widespread in several application domains in data-oriented models. Hence, this gave birth to a new field of study, Scientific Machine Learning (SciML). This field is concerned with the proper application of an ML model, considering all peculiarities of a given domain. Several concerns have risen as ML becomes more prevalent in engineering applications (CHUANG; KEISER, 2018; GAIKWAD et al., 2020; RACKAUCKAS et al., 2020; NOGUEIRA et al., 2022).

One issue that can be highlighted is developing uncertainty-assessed machine learning models for process engineering (MOOS et al., 2022). In the literature, it is possible to find recent works concerned with developing uncertainty-aware models (DAS; KHANNA, 2021; PSAROS et al., 2022), called robust learning (MOOS et al., 2022).

Uncertainty is an important topic, as developing robust models is essential for applying an ML in a real-case scenario. Most application domains have associated uncertainties caused by corrupted data, measurement noise, redundancies, and instrument uncertainties. When these issues are not considered, the ML tools may perform poorly and become inadequate. This issue might create skepticism in the general scientific community toward ML tools. Hence, robust learning plays an essential role in SciML literature (MOOS et al., 2022).

Despite its increasing interest, there still needs to be more algorithms that efficiently cope with the epistemic and aleatory uncertainties in real case scenarios (LI, 2018; DAS; KHANNA, 2021). For instance, Gal e Ghahramani (2015) presents a seminal work regarding this topic. The authors proposed evaluating reinforcement learning models using a Bayesian approximation technique. The referred article points toward issues still needing to be addressed in this field.

Fully understanding the uncertainty of ML models is still a complex issue, as it means evaluating the uncertainty of predictions. Abdar et al. (2021) presents a comprehensive review on this topic and reinforces the ideal of an increasing necessity for further developments. This thesis also contributes to this field by proposing a strategy for developing uncertainty-aware soft sensors based on Deep Learning architecture. However, there will be one need for more in the literature: developing methods that simultaneously evaluate the different uncertainty components involved in an ML model identification. In other words, it is necessary to understand how the number of neurons, activation functions, optimization algorithms, regularization, and other techniques impact the artificial intelligence model uncertainty.

While identifying an ML model, there are several perspectives to be considered. For instance, Abdar et al. (2021) proposed three primary uncertainties related to an ML application: the causality of the models, the sensitivity to data corruption or imperfection, and the computational effort. As mentioned, several methodologies in the literature address one of these listed points. However, the literature needs methods to address intersection applications in the same framework.

#### **1.4.1 Electric submersible pumps**

In the context of electric submersible pumps (ESP) for artificial lift in oil production, validated and reliable models help to develop control schemes and optimize the operating system. A proposal for the dynamic modeling of ESP systems wells first emerged in the work of Batista (2009). In that work, the model has a hybrid construction based on empirical correlations for the pressure gradient and mass balance in the annulus of a well to obtain the liquid level. Barbosa (2011) proposes including in the simulator of Batista (2009) the model of several components of the ESP system, such as variable-speed drive, electric cables, transformers, and electric motors. Further, Barbosa (2011)



proposes modeling the heat exchange between the fluids and the motor under the natural convection hypothesis. The main limitation of these studies is the limited validation of the system: for example, Barbosa (2011) presents only a stationary validation. In contrast, the model was used for multiple controllers testing by Costa (2012), Costa et al. (2013), Filho e Maitelli (2017).

Rønning (2011) presents a simple dynamical model for the ESP system in a different approach. The model is an algebraic differential equation system obtained from the system's mass and moment balance under the hypothesis of two control volume systems, homogeneous fluid, and isotherm conditions. Outputs of the model are the pump's intake pressure, wellhead pressure, and average flow rate. Inputs are the ESP speed, choke opening factor, reservoir pressure, and manifold pressure. Furthermore, Rønning (2011) includes the pump performance correlations, reservoir inflow, and wellhead choke valve. This model is more simplified than the one proposed by Batista (2009) since it does not depend on several correlations with the surface equipment or motor model and is suitable for control design (KRISHNAMOORTHY et al., 2016).

The Rønning (2011) model's primary objective is to provide a simulator to investigate the ESP intake pressure control under its operational constraints. Several papers have been presented using the same model to provide control, state estimation, and optimization solutions for the ESP system: Sharma e Glemmestad (2013), Binder, Kufoalor e Pavlov (2014), Pavlov et al. (2014), Binder, Pavlov e Johansen (2015), Krishnamoorthy et al. (2016), Binder, Johansen e Imsland (2019), Delou et al. (2019).

Using another application of ESP systems, Kullick e Hackl (2017) presents a complex model of the use of ESPs to lift geothermal fluid. The authors derive a model of a completely homogeneous and isothermal ESP system built with three subsystems to model the mechanical, electrical, and hydraulic systems. All subsystem models are obtained using phenomenological equations, including the same momentum and mass balance for the hydraulic subsystem as Rønning (2011). The model proposed by Kullick e Hackl (2017) has 27 state variables. It is suitable for general knowledge about the system and has already been used in digital twin (DT) applications, as seen in Lastra (2019). However, the complete model increases the difficulty of building a control application. A solution for this issue is to use a simple mechanical model equivalent to that proposed in Rønning (2011).

In contrast, to obtain validated models, it is necessary to know the influence of all the equation system parameters and how they change over the operation, including the time-driven modification of the fluids' properties and the equipment's behavior. Accordingly, Sharma e Glemmestad (2014) presents an uncertainty and sensitivity analysis of an ESP model system. The authors propose an uncertainty analysis to measure input parameters' influence—such as production index (PI), reservoir pressure, and water cut—in the uncertainty of four wells' production flow in a hypothetical field. The sensitivity analysis evaluates which parameter impacts the system's behavior. In its turn, the uncertainty analysis presented by Sharma e Glemmestad (2014) uses the Monte Carlo method to sort the input parameters under the hypothesis of Gaussian distribution with an arbitrary  $\pm 10\%$  of uncertainty range.

The Sharma e Glemmestad (2014) method's main limitation is that it does not ad-

dress the evaluation of the parameters' value. The authors assume that the parameters reservoir, pump, and choke valve have nominal values and normal distributions and do not assess whether these parameters correspond to the process's reality. Likewise, the work of Rønning (2011) and Pavlov et al. (2014) suppose fixed parameters equal to the manufacturer's values when the equipment was new. Historically, each of these parameters has been estimated separately. For instance, several methods are present in the literature to obtain the ESP system's performance curves, including the traditional empirical correlations, mechanical modeling, and new approaches based on computational fluid dynamics (ZHU; ZHANG, 2018). However, recalculating these parameters allows for updating the operating zones of the ESP system. Furthermore, it improves the fit between the experimental data and the model's prediction to ensure it is validated.

One of the control strategy constraints in ESP systems is the operational region envelope (PAVLOV et al., 2014). In the field, the primary source of the envelope is the factory information. The factory provides a performance curve that can be interpreted as an ideal region of operation. According to Takacs (2018), these curves are obtained experimentally and under controlled conditions based on **API RP 11S2**. Once in operation, the fluid changes depending on process conditions. In this way, the stages suffer wear and, as a consequence, the performance of the pumps.

One way to update the ESP models' parameters is by the Bayesian method proposed by Gamerman e Lopes (2006). Based on existing information parameters, the Bayesian method built the system's posterior probability density function (PDF). Additionally, this approach allows for the model's validation since the uncertainty of the model's prediction is obtained by propagating a random sample taken from the PDF. The model validation is made by comparing the model prediction and experimental data to analyze the overlapping between the experimental and model confidence bounds.

Regarding the problem of modeling ESP systems, the focus has been on finding linear or nonlinear phenomenological dynamic models (RØNNING, 2011; PAVLOV et al., 2014; KRISHNAMOORTHY et al., 2016; COSTA et al., 2021). These models are Stiff nonlinear ODEs, which are costly and possibly unstable to simulate and can be troublesome for real-time applications, e.g., control and real-time optimization (RTO). Hence, the literature lacks studies regarding the application of AI to ESP process modeling to avoid the complications of Stiff ODE solving and instability.

Another approach option is using data-driven models. These models, especially artificial neural networks (ANNs), are outstanding science, engineering, and technology techniques. Specifically for oil production applications with electric submersible pumps (ESP) systems, research has focused on the use of artificial intelligence in various applications, including surveillance, efficiency improvement, modeling, control, optimization, and others (MOHAMMADZAHERI et al., 2020; ALAMU et al., 2020; CHOUBEY; KARMAKAR, 2021; LI et al., 2021; ABDURAKIPOV, 2021). Also, AI applications have been used to address issues related to predicting the mechanical performance of pumps (MOHAMMADZAHERI et al., 2020).

The first step while developing an AI model for an ESP system is to consider the data structure, the predictor, where the nonlinear approximator will be applied. Thus, Haykin (1999), Géron (2019), Nelles (2020) presents several learning models and points

out that nonlinear autoregressive predictors with exogenous inputs (NARX) are suitable for representing dynamic systems time series. Therefore, this work proposes to group the operating variables of the ESP system through the NARX predictor architecture, hence compiling a dynamic predictor that emulates mechanistic models such as the models of Rønning (2011), Pavlov et al. (2014), Krishnamoorthy et al. (2016) and Costa et al. (2021). Then, the ANN model can be trained using the predictor structure.

The presented scenario indicates that the ESP literature suggests a gap in evaluating the nonlinear model parameters in an integrated manner and creating AI models. Besides, the proposed models lack validation through experimental data, including systematic uncertainty analysis. To fulfill those literature gaps, Chapter 3 presents a representative and validated ESP model that considers uncertainty analysis. The model is based on phenomenological balance equations and depends on parameters representing reservoir, choke, and pump performance. This thesis also proposes to use the information of the process variables to update the envelope region and obtain the parameters with the appropriate propagation of measurement uncertainty using Bayesian inference. Experimental data from an ESP prototype is used in the parameter estimation and model validation. The uncertainty assessment is performed on the experimental data, parameters, and model prediction. The ESP system's performance pump parameters delimit an envelope pump zone (see Pavlov et al. (2014)), so the proposed method allows updating these envelope pump zones.

The AI approach is shown in Chapter 4. The proposal is to introduce a novel perspective for modeling electric submersible pump (ESP) systems, incorporating uncertainty assessment in the ANN model. The empirical modeling of an ESP process is established by leveraging a combination of experimental and synthetic data. This primary contribution lies in presenting an approach to build and validate machine-learning models tailored explicitly for ESP systems. The critical concept involves calculating the uncertainties associated with an experimentally validated nonlinear model proposed by Costa et al. (2021) and propagating these uncertainties to Machine Learning-based models. Some basic ideas are shown in advance in Chapter 2 and revisited in Chapter 4.

On the other hand, the literature has presented several advances in evaluating uncertainty in AI models. Menezes et al. (2018) employ the uncertainty propagation law of distributions proposed by supplement 1 to the Guide to the expression of uncertainty in measurement (GUM-S1) BIPM et al. (2008) to evaluate the uncertainty of static models of ANN. Abdar et al. (2021) reviews uncertainty analysis methods for deep learning models. Among these techniques, the methods based on Bayesian inference and Monte Carlo simulations stand out. Thus, this work will use the Monte Carlo simulation method to evaluate the uncertainty of the ANN models proposed for the ESP system.

This thesis addresses the open issue of empirical modeling of the ESP system coupled with uncertainty prediction. It contributes by proposing a robust methodology to compute uncertainties in a validated nonlinear model presented in Chapter 2 and propagating these uncertainties to Machine Learning-based models. The approach, Markov chain Monte Carlo (MCMC) simulations, generates a set of ANN models trained based on the uncertainties from the nonlinear model. By incorporating uncertainty assessment following the Bayesian paradigm, the proposed approach enhances the reliability and ro-

bustness of Machine Learning models, making notable strides in ESP system modeling and uncertainty prediction.

#### 1.4.2 Pressure swing absorption

Pressure swing adsorption (PSA) is widely employed for separating and purifying gas phases, such as in the catalytic reforming industry. It is often used to purify synthesis gas and prepare it for the Fischer-Tropsch process (RIBEIRO; SANTOS; RODRIGUES, 2011). PSA operates by varying the pressure of the adsorbent bed, enabling selective adsorption and desorption of targeted components at specific process stages. As a result, the separated elements are present in the process discharge, presenting high purity levels. However, this leads to a cyclic behavior without a steady state, which makes it hard to operate due to the unit's complexity, such as non-continuous behavior at the local level, necessary switches at the system boundary conditions, and periodic and persistent changes at valves and pressure controllers (RIBEIRO; SANTOS; RODRIGUES, 2011; NOGUEIRA et al., 2020a).

PSA systems are typically modeled using mass, energy, and momentum balance equations, resulting in partial and algebraic differential equation (PDAE) models Regufe et al. (2015), Rebello et al. (2022b). Solving these complex models demands extensive computational resources. Moreover, the process's reliable operation requires consistent models capable of tracking the system's nonlinearities while providing dependable information for decision-making (NOGUEIRA et al., 2020a; SILVA et al., 2020; MARTINS et al., 2021). Thus, PSA systems are ideal and challenging candidates for AI-based modeling (NOGUEIRA et al., 2022). AI-based models become particularly critical when designed for monitoring, optimization, and control applications, as the recursive numerical solution of PDAE problems can be computationally intensive. However, identifying a reliable AI model for a PSA is challenging due to the process's unique characteristics (NOGUEIRA et al., 2020a).

Regarding the use of AI in PSA systems, several studies have addressed AI modeling, acknowledging the challenges associated with these tasks. For example, Oliveira et al. (2021) presented a preliminary approach to applying deep learning for the dynamic modeling of a PSA unit. This study demonstrated that ANN can model periodic dynamics with relatively high precision and perform significantly better than traditional approaches. In Subraveti et al. (2022), the authors proposed using a physics-informed neural network for modeling a PSA unit, illustrating the power of the AI approach in modeling PSA systems. Another study, using a Bayesian approach, Costa et al. (2022), developed an uncertainty analysis of AI models for a PSA process. This work was done for simulation purposes, neglecting the peculiarities of employing such models and their uncertainties. However, to date, all works related to AI applications for PSA modeling focus on simulation without implementing online applications involving real-time information exchange between the system and models, which would constitute a digital twin (DT).

DT can be crucial in monitoring and predicting the PSA process, as it allows process digitization and reduces the need for direct plant interventions. Optimization algorithms

and controls can be tested and validated within the DT before being implemented in real-time plant operations (MARTINS et al., 2021).

Furthermore, the uncertainty identification of AI models is challenging and has yet to be applied in building DT for PSA units (ABDAR et al., 2021; REBELLO et al., 2022a). Also, the characterization of AI uncertainty creates a trade-off between reliability and computational effort, as the final result comprises a hyperspace of thousands of model parameters, each with a probability of accurately representing the system (ABDAR et al., 2021). This distribution of probabilities must be addressed when integrating such information into a DT. Therefore, it is imperative to develop innovative strategies incorporating the rigor of proper uncertainty identification while accommodating this information within an online monitoring system, ensuring computational feasibility and reliability for the DT.

Assessing uncertainty is crucial for effectively implementing DTs and AI applications in engineering (ABDAR et al., 2021). These AI tools, designed to enhance decision-making, monitoring, diagnostics, and predictive maintenance, must account for uncertainties in data, models, and decisions to ensure reliability, safety, and effectiveness. Data uncertainties may arise from measurement errors, sensor inaccuracies, and data incompleteness, while model uncertainties can originate from structural and parametric sources, potentially leading to inaccuracies in predictions and simulations. As DT and AI technologies evolve, further research and development in uncertainty assessment methodologies will be essential for advancing the application and accuracy of DT systems.

An important aspect that increases the uncertainty and needs to be considered when operating a PSA system is the inevitable degradation of the adsorbent over time. This degradation alters the process behavior, and models usually built to predict the system's performance will, over time, drift away from its initial state. Hence, continuous monitoring and learning have become essential tools in PSA systems, and this study aims to address this issue. This work simulates a PSA operation where adsorbent degradation is observed. We develop a DT that recognizes its prediction drift and adjusts to the new system conditions, leveraging online learning to respond effectively to the changing environment. This approach allows the model to evolve with the system, improving its reliability and performance.

To contribute to the PSA-DT uncertainty assessment, Chapter 5 proposes a DT incorporating online learning and uncertainty assessment for the PSA system, addressing the challenges of uncertainty identification and computational feasibility. In this way, the methods proposed in Chapter 2, 3 and 4 were adapted and applied to build a robust DT to cyclic separation systems, in that case, the PSA processes. The proposed strategy for digitizing such processes and building an uncertainty-aware DT represents an innovative approach that can advance the state-of-the-art in this field. Moreover, the contributions extend beyond cyclic processes, providing valuable insights into artificial intelligence applications and DT.

By leveraging AI technologies, we can enhance the understanding of complex systems and optimize their performance, thus benefiting a wide range of industries and applications. Then, the development of the DT for the syngas purification system comprises one of the contributions of this thesis. This DT is uncertainty-aware and reliable while

continuously updating itself through online learning. Consequently, online learning is a crucial component of the DT. A novel feedback tracker is incorporated within the DT, functioning as a classical feedback controller to maintain the DTs accurate representation of the PSA system. This innovative approach enables the DT to utilize online learning when needed, precisely tracking and adapting to the system's behavior over time.

The proposed methodology application creates a robust and adaptable DT for the PSA system, capable of accurately representing the system's dynamics and assessing the prediction uncertainties of the ANN model. Through a combination of offline training and online learning, the DT can continuously improve its performance, support an optimal operation of the PSA system, and facilitate informed decision-making for enhanced process control and optimization.

## 1.5 BASIC CONCEPTS

This section will provide some insights into the methods used to build the methodology proposed in this thesis.

### 1.5.1 Parameter estimation and model uncertainty assessment

The construction of mathematical models for system representation primarily relies on parameter estimation. Generally, the models are assumed to have a predefined structure or form and associated parameters that determine how well the model fits the experimental data. In this context, parameter estimation is one of the fundamental tools in system modeling.

The modeling and parameter estimation literature sets several guidelines and assumptions for developing and validating a model against experimental data. In general, a mathematical model is defined as an attempt to explain reality (SCHWAAB; PINTO, 2007), but there will always be uncertainty associated with this model. However, the Perfect Model Hypothesis is a common yet idealized hypothesis associated with model building. Under this hypothesis, it is considered that the model's structure is correct and that it can precisely describe the relationships between process variables, even if its parameters are unknown. Additionally, it is considered that any discrepancies between the model fit and the observed experimental data arise from experimental deviations or uncertainties (SCHWAAB; PINTO, 2007).

Regarding the use of experimental data, it is also necessary to establish the Well-Performed Experiment Hypothesis (SCHWAAB; PINTO, 2007). Under this assumption, the experimental data collected from an experiment are considered reliable and free of any trend or bias in experimental error. In this scenario, the expected average value for the experiment's result converges to the mean value calculated with the model (SCHWAAB; PINTO, 2007).

The method of maximum likelihood, extensively discussed in Bard (1974), Schwaab e

Pinto (2007), posits that a parametric model can be generically defined by:

$$\begin{aligned} y_1^m(x, \alpha) &= f_1(x, \alpha) \\ y_2^m(x, \alpha) &= f_2(x, \alpha) \\ &\vdots \\ y_{N_f}^m(x, \alpha) &= f_{N_f}(x, \alpha) \end{aligned} \quad (1.1)$$

in which  $\alpha_p^\top = [\alpha_1, \alpha_2, \dots, \alpha_{N_p}]$  are the parameters associated with the  $N_f$  nonlinear transformation functions of the  $N_x$  independent variables. The principle of the method is the maximization of the likelihood function or minimization of a least squares objective function defined by:

$$F_{\text{obj}} = \sum_{i=1}^{N_e} \sum_{j=1}^{N_f} \left( \frac{y_{ij}^e - f_j(x_i^e, \alpha)}{\sigma_{ij}^2} \right)^2 \quad (1.2)$$

In which  $N_e$  is the number of experiments and the exponent  $e$  indicates experimental information. This function is weighted by the covariance among the parameters defined by  $\sigma_{ij}^2$ .

The solution to the problem can be achieved by calculating the roots of the system formed by the gradient of Equation (1.2) equal to zero, in symbols:

$$\nabla_\alpha F_{\text{obj}} = \mathbf{0} \quad (1.3)$$

Writing the gradient explicitly as a function of Equation (1.2) obtains:

$$\nabla_\alpha F_{\text{obj}} = \left[ \frac{\partial F_{\text{obj}}}{\partial \alpha_1}, \frac{\partial F_{\text{obj}}}{\partial \alpha_2}, \dots, \frac{\partial F_{\text{obj}}}{\partial \alpha_{N_p}} \right]^\top = \mathbf{0} \quad (1.4)$$

in which each derivative is given by:

$$\frac{\partial F_{\text{obj}}}{\partial \alpha_p} = 2 \sum_{i=1}^{N_e} \sum_{j=1}^{N_f} \left( \frac{y_{ij}^e - f_j(x_i^e, \alpha)}{\sigma_{ij}^2} \right) \left( -\frac{\partial f_j}{\partial \alpha_p}(x_i^e, \alpha) \right) = 0, \quad p = 1 \dots N_p \quad (1.5)$$

Given the optimum parameters values  $\alpha_p^{*\top}$ , it is necessary to obtain the covariance matrix to assess the uncertainty of those parameters. Under the hypotheses of uncorrelated experimental measurement of the dependent and independent variables, the covariance matrix can be defined as:

$$V_\alpha = E\{\Delta\alpha\Delta\alpha^\top\} = H_\alpha^{-1} [G_y V_y G_y^\top + G_x V_x G_x^\top] H_\alpha^{-1} \quad (1.6)$$

in which  $V_x$  and  $V_y$  are the covariance matrix of the experimental deviations of the independent and dependent variables. Also,  $H_\alpha$  is the Hessian matrix,  $G_x$  and  $G_y$  are auxiliary derivatives matrix given by:

$$G_y = \frac{\partial (\nabla_\alpha F_{\text{obj}}(y^e, x^e, \alpha))}{\partial y^e} = \begin{bmatrix} \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial y_{1,1}^e} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial y_{1,2}^e} & \dots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial y_{N_e, N_f}^e} \\ \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial y_{1,1}^e} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial y_{1,2}^e} & \dots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial y_{N_e, N_f}^e} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial y_{1,1}^e} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial y_{1,2}^e} & \dots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial y_{N_e, N_f}^e} \end{bmatrix} \quad (1.7)$$

$$G_x = \frac{\partial (\nabla_\alpha F_{\text{obj}}(y^e, x^e, \alpha))}{\partial x^e} = \begin{bmatrix} \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial x_{1,1}^e} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial x_{1,2}^e} & \cdots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial x_{N_e, N_x}^e} \\ \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial x_{1,1}^e} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial x_{1,2}^e} & \cdots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial x_{N_e, N_x}^e} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial x_{1,1}^e} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial x_{1,2}^e} & \cdots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial x_{N_e, N_x}^e} \end{bmatrix} \quad (1.8)$$

$$H_\alpha = \frac{\partial (\nabla_\alpha F_{\text{obj}}(y^e, x^e, \alpha))}{\partial \alpha} = \begin{bmatrix} \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial \alpha_1} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial \alpha_2} & \cdots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_1 \partial \alpha_{N_p}} \\ \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial \alpha_1} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial \alpha_2} & \cdots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_2 \partial \alpha_{N_p}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial \alpha_1} & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial \alpha_2} & \cdots & \frac{\partial^2 F_{\text{obj}}}{\partial \alpha_{N_p} \partial \alpha_{N_p}} \end{bmatrix} \quad (1.9)$$

The principal challenge in addressing the estimation problem via the maximum likelihood approach is the computation of derivatives for Equations (1.4) to (1.9). Bard (1974) examines the application of this method to dynamic models and notes that under such conditions, derivatives cannot be obtained analytically, necessitating a numerical solution. Consequently, when dealing with stiff or numerically unstable models, the computed derivatives may cause significant fluctuations in the estimated model uncertainty, leading to results not reflecting actual conditions. Thus, parameter estimation for intricate dynamic models presents a substantial theoretical and computational challenge.

**1.5.1.1 Bayesian inference** Conversely, parameter estimation techniques that utilize Bayesian inference bypass the need for analytical or numerical derivative computation to ascertain parameters and their associated uncertainty. These experimental-based methods emerge as a practical alternative for determining the uncertainty in complex dynamic models (GELMAN, 2015).

Bayesian inference is based on obtaining the joint probability distribution of  $\boldsymbol{\theta}$  parameters given measurements of  $y$ . In this context, Bayes' rule establishes that:

$$g(\boldsymbol{\theta}, y) = g(\boldsymbol{\theta})g(y|\boldsymbol{\theta}) \quad (1.10)$$

in which  $g(\boldsymbol{\theta}, y)$  is the joint distribution,  $g(\boldsymbol{\theta})$  is the prior distribution and  $g(y|\boldsymbol{\theta})$  is the sampling distribution, or data distribution. In Equation (1.10),  $g(y|\boldsymbol{\theta})$  is the posterior density that, from Bayes' theorem, can be written as:

$$g(\boldsymbol{\theta}|y) = \frac{g(y|\boldsymbol{\theta})g(\boldsymbol{\theta})}{g(y)} \quad (1.11)$$

in which  $g(y)$  is the marginal distribution of  $y$ , under the hypothesis of  $y$  is supposed to be unknown but observable, is given by:

$$g(y) = \int g(y, \boldsymbol{\theta})d\boldsymbol{\theta} = \int g(y|\boldsymbol{\theta})g(\boldsymbol{\theta})d\boldsymbol{\theta}. \quad (1.12)$$



As given by Gamerman e Lopes (2006), the sampling distribution can be defined as the likelihood function, then,  $L(\boldsymbol{\theta}) = g(y, \boldsymbol{\theta})$ . This function provides the probability of each parameter value  $\theta$  having one observed  $\tilde{y}$ . Gelman (2015) also points out the likelihood principle that is established that if two probability models, given by  $g(y|\boldsymbol{\theta})$ , with the same data, have the same likelihood function value, this will bring to the same parameters  $\boldsymbol{\theta}$ .

Therefore, the posterior density is proportional to the product between the prior function and the likelihood, in symbols:

$$g(\boldsymbol{\theta}|y) \propto L(\boldsymbol{\theta})g(\boldsymbol{\theta}) \quad (1.13)$$

In Equation (1.13), the concept of prior and posterior is relative and conditioned on the information available at a given time. This means that in Equation (1.13), proportionality is maintained before observations are made. However, it is considered that the Bayesian inference process indicates a constant update so that, after observing  $y$ , a subsequent one can be calculated. Following this logic, if the likelihood function concerning the  $\boldsymbol{\theta}$  parameters is updated at some point, this will generate a new update of the subsequent one by applying Bayes' theorem.

The predictive approach to inference can be used to estimate parameters, even if the parameters are not observed. Gamerman e Lopes (2006) establishes that for a case of multivariable parameters  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{NP})'$ , the conditional and marginal distributions of each component of  $\theta_i$  can be obtained from the joint distribution  $g(\boldsymbol{\theta})$ . Thus, the posterior distribution of  $\boldsymbol{\theta}$  is:

$$g(\boldsymbol{\theta}|y) = \int g(\theta_1, \dots, \theta_d) d\boldsymbol{\theta}_{-i} \quad (1.14)$$

in which  $\boldsymbol{\theta}_{-i} = (\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_d)$  is the vector  $\boldsymbol{\theta}$  without the  $i$ th component,  $i = 1, \dots, NP$ .

The solution of Equation 1.14 is obtained The solution to Equation 1.14 can be obtained through several methods, such as Laplace approximation, quadrature approximation, Monte Carlo integration, and resampling techniques (GAMERMAN; LOPES, 2006). Gamerman e Lopes (2006) also points out that Monte Carlo techniques are the option when the problem is complex or the dimension is high. This thesis will apply the Markov chains simulations to solve the inference problem.

The Markov chains simulations can be performed through the Metropolis-Hastings algorithm, which can be defined as (GAMERMAN; LOPES, 2006; GELMAN, 2015):

1. Initialize the parameters of  $\theta^0$  from a starting distribution  $p_0(\theta)$ , and set the counter  $j = 1$ .
2. Move the chain to a new value in  $j = 2, \dots, N_{samples}$ .
3. Draw a point from the proposal distribution  $J_t(\theta^*|\theta^{j-1})$ .
4. Calculate the acceptance probability given by the ratio between the sampled and the immediate anterior point:  $r = \frac{g(\theta^*|y)}{g(\theta^{j-1}|y)}$ .

5. Evaluate the new point:  $\theta^t = \begin{cases} \theta^* & \text{with acceptance probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$
6. Move the chain to a new value in  $j = j + 1$  and return to 2 until converge.

In step 3,  $J_t(\theta^*|\theta^{j-1})$  also can be called jump distribution or kernel distribution (GAMERMAN; LOPES, 2006; GELMAN, 2015).

Step 5 requires a sample of an independent quantity  $u$  from a uniform distribution. If  $u \leq \min(r, 1)$ , the transition is accepted and  $\theta^t = \theta^*$ . However, if  $u > \min(r, 1)$ , the transition is not accepted and  $\theta^t = \theta^{t-1}$ . In any case, this comparison is counted as an iteration in the algorithm. This algorithm uses the proposal distribution to evaluate if the transition can be confirmed. It is also important to point out that  $J_t(\theta^*|\theta^{j-1})$  must be symmetric so  $J_t(\theta^a|\theta^b) = J_t(\theta^b|\theta^a), \forall(\theta^a, \theta^b)$  and  $j$ .

Following previous ideas, many algorithms have already been developed to perform Monte Carlo simulations and build Markov chains. Many algorithms have included mechanisms to facilitate convergence and numerical stability. One of these algorithms is the DRAM presented by Haario et al. (2006a), which includes the delayed-rejection mechanism to improve the convergence process.

The adequate number of draws collected from the proposal and the number of iterations can be calculated based on the chain's variance (GELMAN, 2015; ROY, 2020). This parameter is essential to define the chain's convergence. Also, it is crucial to determine some draws as burn-in so those samples are used to update the chain variation, but they are discarded at the end to ensure that the chain is not dependent on the initial samples. On the other hand, the literature already defined some metrics to test if the chain converged (GEWEKE, 1992; ROY, 2020).

Geweke (1992) propose the Z-test equality of means in which the standard error is calculated and used to find the autocorrelation of the samples. In symbols:

$$Z_n = \frac{(\bar{g}_{n_A} - \bar{g}_{n_B})}{\sqrt{\hat{S}_g(0)/n_A + \hat{S}_g(0)/n_B}} \quad (1.15)$$

in which  $Z$  is the Geweke's score between two parts of the chain,  $n_A$  and  $n_B$ .  $\hat{S}_g(0)/n_A$  and  $\hat{S}_g(0)/n_B$  are the asymptotic variance of the time averages of the  $n_A$  and  $n_B$  observations in the chain  $\bar{g}_{n_A}$  and  $\bar{g}_{n_B}$ .

Geweke's parameter is analyzed by selecting ranges from the chains and checking whether their values are close to unity. Values close to one indicate that the chain has converged. Small values, close to zero or less than a certain arbitrary threshold, may indicate that the chain has not converged and needs more samples or modifications in one of the distributions involved.

### 1.5.2 Neural networks for system identification

The construction of neural networks is fundamentally inspired by the cellular brain biology observed in nature, prevalent in most living organisms. The architecture of cellular neurons laid the groundwork for the development of the artificial neuron. Similarly, the

connections between biological neurons and synapses inspired the computational flow of artificial neural networks (ANNs) (Hands on 2022). Based on the concept of synapses and brain activation observed in living beings, artificial neural networks are stimulated through inputs, and specific areas are activated based on the logical conditions computed throughout the network.

The basic structure of a perceptron unit is illustrated in Figure 1.1. In this structure, information flow occurs from left to right. Accordingly, the inputs are multiplied by the weights  $w$  associated with each input. These values are then summed with the bias  $b$ . Following this operation, the activation function is applied with a threshold represented by  $T$ , and the result of this operation yields the output value predicted by the neuron. Suppose this perceptron structure is interconnected with other subsequent or parallel perceptions, as in Figure 1.2. In that case, it forms the structure known as a dense layer neural network, a fully connected neural network, multi layer perceptron (MLP), or, often referred to in many texts, an Artificial Neural Network (ANN). This term will be adopted in this work.

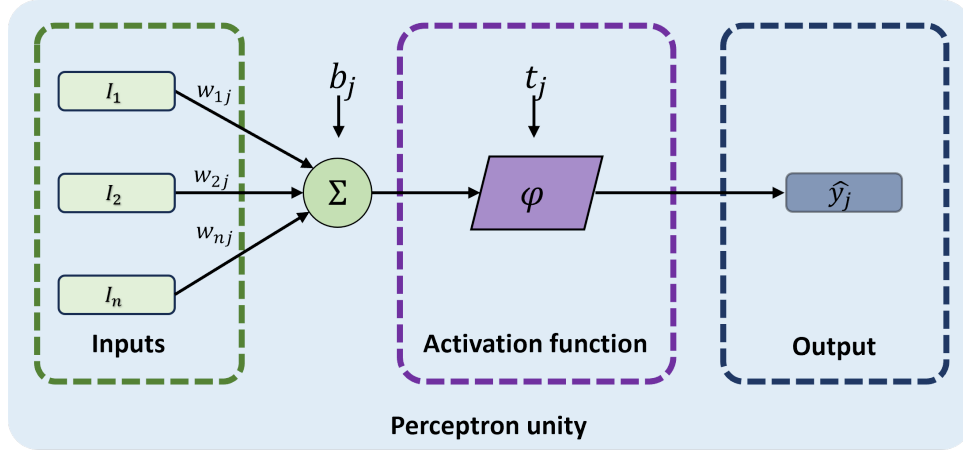


Figure 1.1: Perceptron scheme.

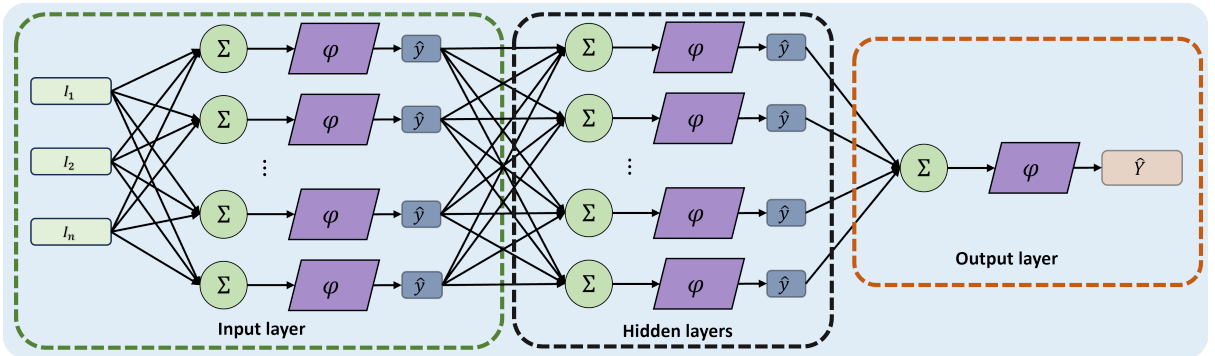


Figure 1.2: Multilayer perceptron.

In both scenarios,  $I_{i,...,n}$  represents the system's inputs, which can be exogenous inputs

or past values of the predicted variable, known as regressors.  $w_{1j,2j,\dots,nj}$  denotes the  $ij$ -th neuron weights,  $b_j$  represents the  $j$ -th neuron biases, and  $t_i$  are the  $j$ -th neuron activation function thresholds.  $\hat{y}_j$  are  $j$ -th the predicted values for each neuron, and  $\hat{Y}$  is the value predicted by the MLP.

**1.5.2.1 Definitions** In the MLP structure, the activation function holds a central focus and has been extensively discussed in the literature. This profound discussion significantly impacts how networks are constructed and their predictions behave. Likewise, the literature has proposed several different approaches to activation functions so that the choice of the activation function becomes an element to be obtained during the optimization of hyperparameters. Several algorithms are available as libraries for searching hyperparameters, including random search, Bayesian optimization, and hyperband, which are examples of algorithms that deal with these choices (BISCHL et al., 2021). This choice largely depends on the problem being addressed or the type of data being modeled. Table 1.1 provides a general overview of the literature’s most commonly encountered activation functions. Mathematically, these functions aim to serve as a gate that amplifies or diminishes the output of a neuron, thereby activating or deactivating its response.

Table 1.1: Examples of activation functions.

Name	Expression
Absolute value	$id(x) = x$
Sigmoid	$\alpha(x) = \frac{1}{1+e^x}$
Cosine	$cos(x)$
Hiperbolic tangent	$tanh(x) = \frac{2}{1+e^{-2x}} - 1$
Rectified Linear Unit	$ReLU(x) = \max(0, x)$
Exponential Linear Unit	$ELU(x) = \max(0, x) + \min(0, \alpha(e^x - 1))$
Scaled Exponential Linear Unit	$SELU(x) = \gamma \cdot \max(0, x) + \min(0, \alpha(e^x - 1))$
Heaviside	$H(x) \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x \leq 0 \end{cases}$

Hyperparameters, such as the number of layers, neurons, and initial learning rate, play an essential role in current artificial intelligence models so that AI models become highly configurable, and the choice of hyperparameters can become complex and dependent on the data you want to model. In this sense, hyperparameter algorithms also act on this choice. Thus, their primary objective is to optimize hyperparameters automatically, eliminating the need for trial and error, which is time-consuming and can generate biases and errors or be computationally reproducible (BISCHL et al., 2021).

Likewise, hyperparameters interfere with training speed, model complexity, and the volume of data used. In this sense, the hyperparameter search stage, although it also has a high computational cost, allows the reduction of computational effort in the training and use of the model stages and adequate model performance (BISCHL et al., 2021).

Layers organize the typical structure of currently used artificial neural networks. A layer is defined as a set of neurons organized parallel, as represented in Figure 1.2. Sometimes, the structure necessary to characterize the data comprises just one layer of neurons. The network can be called a shallow network in this scenario. On the other hand, some networks are built with several hidden layers and a series of neurons. In this case, it is called a deep neural network (BISCHL et al., 2021), or multi-layer perception.

Another important classification comes from the internal structure of neurons and networks. The literature classifies models concerning the relationship between inputs and outputs. Nelles (2020) Chapter 19 discusses linear and non-linear modeling. It makes a sequential construction starting from the basic linear autoregressive predictor with exogenous inputs (ARX) and expanding to the non-linear models with exogenous inputs (NARX). This discussion also includes the model's ability to predict or simulate the behavior of data. In the case of one-step-ahead prediction, the model is called pure feedforward and the predicted output is given by (NELLES, 2020):

$$\hat{y}(k) = f(u(k-1), u(k-2), y(k-1), y(k-2)). \quad (1.16)$$

In turn, when the model is capable of emulating dynamic behavior and has an internal recurrence, that is, the internal states of the model are automatically fed back, this model is called recurrent. This approach can be represented by:

$$\hat{y}(k) = f(u(k-1), u(k-2), \hat{y}(k-1), \hat{y}(k-2)). \quad (1.17)$$

Several arrangements may be possible, so hybrid structures between feedforward and recurrent can be used for different applications.

Dense layers present a limitation in terms of memory. A solution to this problem emerged with the proposition of the Long-Short-Term Memory (LSTM) unit by (HOCHREITER; SCHMIDHUBER, 1997). This basic unit type can be used as an alternative to the structure of conventional neurons in system simulation applications. This primary cell includes a series of gates defined as short-term and long-term states; in other words, the network has long-term and short-term memory. With a state-oriented structure, the LSTM unit has two sets of states: cell state, which is responsible for long-term memory, and hidden state, which is related to short-term memory, as seen in Figure 1.3.

In the LSTM cell of Figure 1.3, the cell state is responsible for long-term memories. Some memories are discarded during the forget gate operation, but new ones are added in the subsequent step. After adding memories, these long-term states are copied, processed through a tanh function, and filtered by the output gate. This process produces the short-term memory state, known as the hidden state  $h(t)$ . On the other hand, network inputs  $x(t)$  are directly fed into fully-connected layers. Layer 3, featuring the tanh activation function, processes both the inputs and the short-term states. Layers 1, 2, and 4 are responsible for activating or deactivating the gates, as they typically have activation functions scaled between 0 and 1. Thus, the forget gate determines which parts of the long-term terms are erased, the input gate controls which information from layer three is retained in the long-term term, and the output gate controls which information from the states and predictions are read and passed forward (GÉRON, 2019).

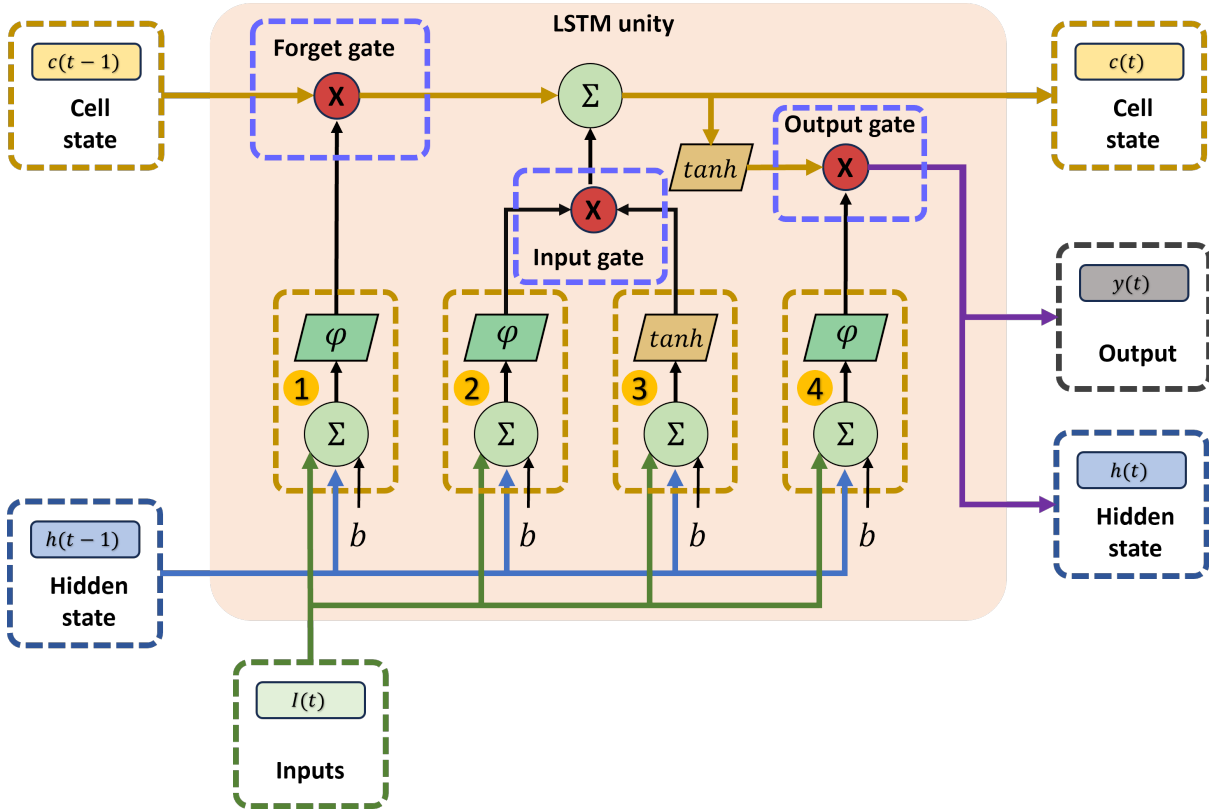


Figure 1.3: Long-Short-Term Memory cell.

Building models with ARX format presents challenges regarding the number of delays necessary for a good output characterization. For models with multiple inputs and single outputs (SISO), He e Asada (1993) established a coefficient calculated by determining the ratio and variance of output values  $y$  and the Euclidean distance between the corresponding input vectors  $x$ , and is given by (HE; ASADA, 1993):

$$q_j^{(m)} = \frac{|\delta y|}{\sqrt{(\delta x_1)^2 + \dots + (\delta x_m)^2}} = \frac{|f_1 \delta x_1 + \dots + f_m \delta x_m|}{\sqrt{(\delta x_1)^2 + \dots + (\delta x_m)^2}} \quad (1.18)$$

in which  $m$  indicates the number of input variables used. This coefficient is fundamental to define the ideal number of delays to be included, calculated through the Lipschitz Index, presented as follows (HE; ASADA, 1993):

$$q^{(n)} = \left( \prod_{k=1}^p \sqrt[n]{q_j(k)} \right)^{\frac{1}{p}} \quad (1.19)$$

This equation can be applied to different numbers of delays and inputs. The indices calculated for each combination can be compared in graphic formats. The straightforward approach is to analyze the index variation between different combinations.

This methodology involves testing different values for the number of delays,  $n$ , and calculating the Lipschitz index for each value tested. The goal is to identify an inflection point where variations in the value of  $n$  no longer lead to significant improvements, thus reducing the ideal number of delays for inputs into the model. This index allows for determining the ideal number of delays for exogenous output variables, quantifying how much input values can affect output values. Hence, an appropriate number of delays, or the correct value of the Lipschitz coefficient, ensures that the NARX model is robust and resistant to small changes in inputs or disturbances.

**1.5.2.2 Loss functions** Neural network training is conducted by selecting the weights and biases for each neuron according to predefined metrics. The most common metrics in the literature originate from the least squares problem, which defines the distance between the data ( $y_i$ ) and the predicted values ( $\hat{y}_i$ ). The primary function is the sum of the squares of the errors and the sum of absolute errors, denoted as SSE and SAE, respectively. In symbols:

$$\text{SSE} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1.20)$$

$$\text{SAE} = \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1.21)$$

$$(1.22)$$

This expression has given rise to several widely-used metrics today, namely, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), also called Euclidean norm, whose expressions are:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1.23)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1.24)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1.25)$$

These metrics are fundamental tools for evaluating the accuracy of neural network models during training. In general, the best-trained values are obtained by finding the minimum value of the distance between the data and predictions. The two most common loss functions used for time series prediction are RMSE and MAE. Some algorithms and libraries already use both during the training (for example, in Keras (CHOLLET et al., 2015), it is possible to define the *loss* function and another one as *monitor function*). Still, the use depends on the problem and needs to be defined carefully (HODSON, 2022).

Therefore, the function that guides the training is called the Loss function. Other options of functions and methods are also presented in the literature to guide the training.

For example, Nelles (2020) categorizes those functions in Bayesian methods, maximum-likelihood, maximum a-posteriori, weighted last-squares, and last-squares methods.

**1.5.2.3 Solving algorithms** Neural network training is carried out iteratively, where values for weights and biases are selected, and the loss function is evaluated. The difference between algorithms originates from how the weights and biases are updated. Among the most common categories are gradient-based algorithms, stochastic algorithms, and nature-inspired algorithms. Generally, deterministic algorithms are guided by the gradient direction or another characteristic of the loss function. Meanwhile, stochastic algorithms randomly select new values, which may or may not be based on previous values and are constrained by search regions. Nature-inspired algorithms choose new values based on the population's best results in earlier iterations (ABDOLRASOL et al., 2021).

On the other hand, more modern algorithms for training networks rely on hybrid constructions. These algorithms aim to leverage the best features of deterministic and stochastic algorithms and are specially developed with a focus on performance, balancing numerical precision, and solution speed. Additionally, the more advanced algorithms are better suited for hardware that supports parallelization and acceleration, such as GPUs and TPUs. A detailed discussion on this topic can be found in Géron (2019) and Abdolrasol et al. (2021).

### 1.5.3 Uncertainty assessment Experimental data

Uncertainty analysis plays a central role in this thesis, analyzing experimental data and predicting the outcomes of constructed models. Uncertainty analysis enhances the understanding of the model's predictive capabilities. Additionally, it allows for the final validation of the models against experimental data and their associated uncertainties in a more objective manner. Two distinct techniques are employed for different purposes. The first is the uncertainty analysis of experimental data, assessed using the guide to the expression of uncertainty in measurement (GUM) method defined on BIPM et al. (2008).

The assessment of data uncertainty may begin with an analysis of the uncertainty of input quantities, as these are obtained directly from the measurement system. Subsequently, the uncertainty of dependent variables is evaluated.

In the context of uncertainty analysis of input variables, defining a function capable of appropriately representing the measurement result of a variable  $X_i$  relates the “i” measurements to the imperfections inherent to the measurement process. BIPM et al. (2008) presents a function that fulfills this role:

$$X_i = Q_i + \sum_{j=1}^N \alpha_{i,j} D_{i,j} \quad (1.26)$$

in which  $Q_i$  is the i-th measurement obtained from the instrument,  $D_i$  is a correction originating from system imperfections, and  $\alpha_i$  is the unit conversion coefficient.

Equation 1.26 aims to estimate the measurement and its associated with it, and the uncertainty of this measurement must be assessed. For this analysis, the ISO Guide



establishes the composition of the uncertainty of an input quantity as a function of type A uncertainty components, which refer to the intrinsic variability of the act of measuring, and type B, originating from supplementary information about the measurement system. This expression can be represented by:

$$u(x_i) = \sqrt{u_A^2 + \left( \sum_{j=1}^n c_j^2 u_{B_j}^2 \right)} \quad (1.27)$$

where  $u(x_i)$  on the left side of the equation is the combined standard uncertainty of the estimate  $x_i$  of the quantity  $X_i$ . The term on the right is formed by the square root of the quadratic sum of type A uncertainty components  $u_A$ , and type B  $u_{B_j}$ , where  $c_j$  is a unit conformity factor.

The type A and B uncertainty components, as well as the values of  $u_A$ ,  $u_{B_j}$ , and  $c_j$  will be calculated in the following sections. However, it is necessary to assess the independence hypothesis among the samples used in the uncertainty analysis beforehand.

**1.5.3.1 Auto correlation** An assumption in the uncertainty analysis using the method proposed by the GUM is the independence between the various measurements of a single variable. Therefore, this hypothesis must be evaluated in the discussion of experimental results. This section discusses this hypothesis and presents the mechanisms to minimize potential interference.

However, process systems have the characteristic of being operated continuously over time. This presents a challenge for analyzing the uncertainty of experimental data according to the GUM, as, strictly speaking, measurements are mutually dependent. At the same time, the main assumption in the standard is the independence between different measurements. One way to ensure the independence of measurements would be to install multiple parallel meters for measuring the same variable. This option is unfeasible; in this case, supplementary mechanisms are used to ensure these hypotheses can be assumed.

The main mechanism employed is to collect samples within a steady state. In this case, the system inputs are held fixed for an extended period to reach a steady state. Upon stabilization, these variables are resampled with temporal spacing to minimize the effect of autocorrelation.

**1.5.3.2 Type A uncertainty evaluation** According to the GUM, the type A standard uncertainty is obtained from the frequency distribution of experimental data. Assuming that the frequency distribution of the measured variable approximates the PDF of a random variable, the first and second statistical moments, that is, the expectation and standard deviation of that measured variable, can be estimated, respectively, through the arithmetic mean and the experimental standard deviation of the individual and independent observations collected. In this scenario, the expressions used to calculate the value of the mean and standard deviation are (BIPM et al., 2008):

$$x_i = \bar{X}_i = \frac{1}{n} \sum_{k=1}^n X_{i,k} \quad (1.28)$$

$$s^2(X_i) = \frac{1}{n-1} \sum_{k=1}^n (X_{i,k} - \bar{X})^2 \quad (1.29)$$

where  $x_i$  is the arithmetic mean of the  $n$  independent observations  $X_{i,k}$  of the measured variable. The experimental standard deviation is obtained through the positive square root of  $s^2(X_i)$  of the same sampled set. Consequently, the type A standard uncertainty ( $u_A$ ) is obtained through:

$$u_A(x_i) = \frac{s(X_i)}{\sqrt{n}}. \quad (1.30)$$

**1.5.3.3 Type B uncertainty evaluation** The Type B uncertainty, the second term on the right side of Equation (2.2), originates from distinct sources and supplementary information about the measurement system and, in this case, includes all the documentation available on the sensor. Precisely, the uncertainties were calculated from the manufacturers' manuals and are associated with i) resolution, ii) accuracy, and iii) drift over time (or instrumental drift). These will be addressed individually in the following.

#### Resolution

To calculate the component related to uncertainty, a rectangular distribution was assumed to be as conservative as possible, and the uncertainty values were calculated based on the expression:

$$u_r = \frac{r_s}{\sqrt{12}} \quad (1.31)$$

where  $r_s$  is the resolution value indicated in the instrument manual, and the term  $\sqrt{12}$  arises from the general expression for a rectangular distribution. In the absence of this information for a specific sensor, it was assumed, hypothetically, that the resolution is to one decimal place.

#### Accuracy

The uncertainty component related to accuracy is calculated based on the information presented by manufacturers who express the accuracy uncertainty as a percentage relative to the maximum value of the measurement.

#### Instrumental drift

Instrumental drift is a linear function of time where manufacturers provide the linearity parameter as a percentage of the maximum value. The calculation of the uncertainty related to instrumental drift  $u_d$  was carried out using the following expression:

$$u_d = \frac{V_{max} \cdot A_c \cdot d_{inst}}{\sqrt{12}} \quad (1.32)$$

where  $V_{max}$  is the maximum measurement value of the instrument,  $A_c$  is the elapsed time since the acquisition of the instrument, assuming that all were acquired new, and  $d_{inst}$  is the instrumental drift indicated in the equipment manual.

**1.5.3.4 Dependent variables** Measuring some variables with high frequency in dynamic systems can be difficult or impossible for many reasons. In these cases, they are calculated based on other variables that have direct measurements. These variables are defined as dependent quantities. Following the nomenclature of the GUM, the uncertainty of these variables is calculated using the general law of uncertainty propagation.

Given one output variable  $z = f(w_1, w_2, \dots, w_N)$  with  $N$  input quantities  $w_1, w_2, \dots, w_N$ , where each  $w_i$  is described by an appropriate probability distribution. The most probable value of this variable is given by the expectancy  $E(w_i) = \mu_i$  and defined by the mean of the calculated values:

$$\bar{z} = \frac{1}{N} \sum_{i=1}^N z_i = \frac{1}{N} \sum_{i=1}^N f(w_{1,i}, w_{2,i}, \dots, w_{N,i}) \quad (1.33)$$

Associated with the expected value, it is possible to find the variance of the output variable, by propagation, as:

$$\sigma_z^2 = \sum_{i=1}^N \left( \frac{\partial f}{\partial w_i} \right)^2 \sigma_i^2 + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left( \frac{\partial f}{\partial w_i} \right) \left( \frac{\partial f}{\partial w_j} \right) \sigma_{ij} \rho_{ij} \quad (1.34)$$

in which  $\sigma_i^2$  is the variance of  $w_i$ ,  $\rho_{ij} = \frac{v(w_i, w_j)}{(\sigma_i \sigma_j)}$  is the correlation coefficient of  $w_i$  and  $w_j$ .

**1.5.3.5 Uncertainty assessment results** The final expression for the uncertainty analysis of experimental data can be presented in various forms (BIPM et al., 2008). The most traditional way is to adopt the expression of uncertainty in the format  $Y = y \pm k u_c$ , where  $Y$  is the measurand,  $y$  is the most likely value (or measurement result),  $k$  is the coverage factor for a chosen coverage probability, and  $u_c$  is the combined standard uncertainty calculated from Equation 1.27. Similarly, uncertainty can also be helpful when presented in the form of relative uncertainty in percentage values, symbols,

$$u_r = \frac{u_c}{|y|} \quad (1.35)$$

**1.5.3.6 Coverage factor** The coverage factor,  $k$ , is a function of the effective degrees of freedom of the combined standard uncertainty. These are calculated using the Welch-Satterthwaite formula (or W-S formula), which is represented by the following expression (BIPM et al., 2008):

$$\frac{u_c^4(y)}{\nu_{eff}} = \sum_{i=1}^N \frac{(c_i u(x_i))^4}{\nu_i} \quad (1.36)$$

This equation allows for considering all sources of uncertainty, both type A and type B. In the case of type A, the degrees of freedom  $\nu_A$  are related to the number of samples collected for the same variable, hence  $\nu_A = n - 1$ , where  $n$  is the sample size.

In turn, each of the type B uncertainty components has its respective degree of freedom and can be determined based on the quality of the information. Assuming a low degree

of freedom means that the information is useful. Still, more confidence in its value is needed, which will impact the final result of the instrumental uncertainty analysis.

With the effective degrees of freedom  $\nu_{eff}$  determined, an appropriate coverage probability value  $p$  is assumed, and the coverage factor  $k$  is calculated based on a Student's t-distribution.

## CHAPTER 2

# A ROBUST LEARNING METHODOLOGY FOR UNCERTAINTY-AWARE SCIENTIFIC MACHINE LEARNING MODELS

Published on: **Costa, E. A.**; Rebello, C. M.; Fontana, M.; Schnitman, L.; Nogueira, I. B. dos R. (2022). A Robust Learning Methodology for Uncertainty-aware Scientific Machine Learning models. Mathematics, 25 December 2022. DOI: <https://doi.org/10.3390/math11010074>

### 2.1 INTRODUCTION

Robust learning is an essential issue in Scientific Machine Learning (SciML). There is an increasing demand for methods that simultaneously consider all the different uncertainty components involved in SciML model identification. This Chapter establishes the general foundations of the concepts that will be applied and developed in the remainder of the thesis. Therefore, hypotheses 1 and 2 presented in Section 1.1 are directly addressed in this chapter. To fulfill that, this chapter proposes a new methodology for uncertainty evaluation of the SciML model that also considers several possible sources of uncertainties involved in the identification process. The considered uncertainties lie in the causality of models and the parameters' sensitivity. Therefore, providing an overall strategy for the uncertainty-aware models in the SciML field is possible. The methodology is validated through a case study, developing a soft sensor for a polymerization reactor. The results demonstrated that the identified soft sensors are robust to uncertainties, corroborating the consistency of the proposed approach.

### 2.2 METHODOLOGY FOR THE MONTE CARLO UNCERTAINTY TRAINING

This paper proposes a general methodology to build models based on artificial intelligence (AI) with uncertainty assessment. This proposed methodology is divided into five steps to obtain the validated machine learning models with uncertainty assessment. Figure 2.1 presents the general scheme of the proposed method as follows.

The first step is using the Markov Chain Monte Carlo (GAMERMAN; LOPES, 2006) method to obtain the uncertainty of the non-linear model parameters that represent the system. Following the methodology, the validated model is used to generate synthetic data. This synthetic data is used to build neural networks in the following methodology steps. With the data, it is possible to define the type and the general architecture of the neural network (Hyperparameter optimization). After finding the best network architecture, the Monte Carlo simulation training data propagates the uncertainty from the non-linear model (i.e., synthetic data) to the AI model. The last step of the proposed methodology is to perform the validation and uncertainty assessment of the trained model.

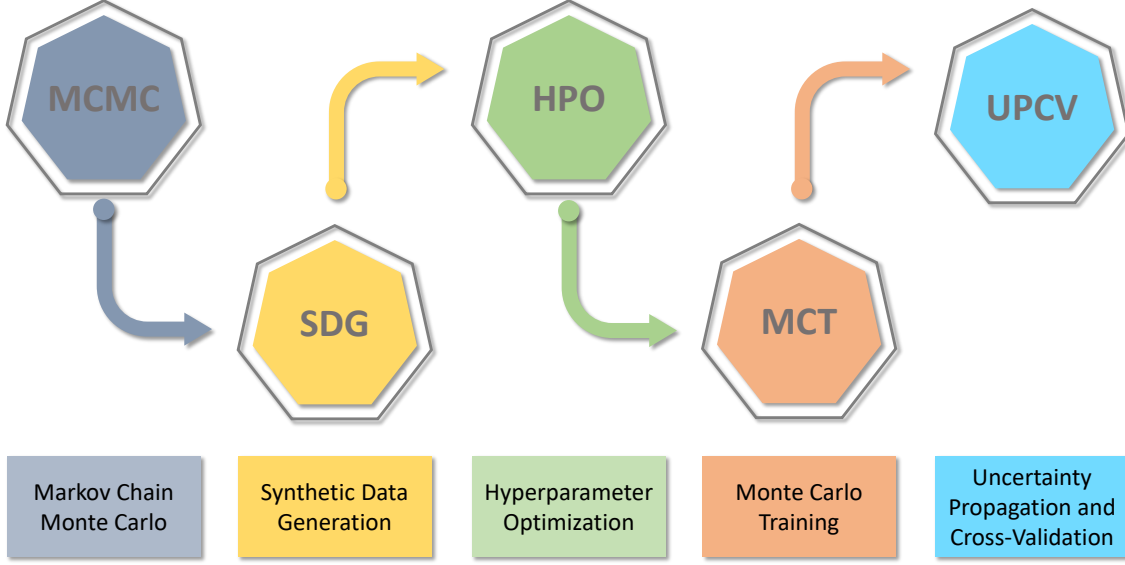


Figure 2.1: General methodology scheme.

In that step, the AI model prediction and simulation are collated with the non-linear model and experimental data if available. Also, the uncertainty assessment is performed in that step. The following subsections will provide specific details about each step of the proposed methodology.

### 2.2.1 Markov chain Monte Carlo

Parameter estimation is a challenge because their choice contributes to the prediction uncertainty of the model. Consider a non-linear dynamic model written as (BARD, 1974):

$$\mathbf{g}(\mathbf{x}, \mathbf{y}, \partial \mathbf{y} / \partial \mathbf{x}, \partial^2 \mathbf{y} / \partial \mathbf{x} \partial \mathbf{x}, \dots, \boldsymbol{\theta}) = \mathbf{0} \quad (2.1)$$

with the respective integrated expression:

$$\mathbf{y} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}, \boldsymbol{\theta}), \quad (2.2)$$

in which  $\mathbf{f}$  is the relationship function of the output vector  $\mathbf{y}$  as a function of the time  $t$ , states  $\mathbf{x}$ , the input vector  $\mathbf{u}$ , and parameters  $\boldsymbol{\theta}$ . In this scenario, if  $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_{np}]$  is a set of  $np$  parameters, and if they have a low predictive probability, the model will not provide a good forecast or adjustment to experimental data (MIGON; GAMERMAN; LOUZADA, 2014). Therefore, knowing the model's parameters' probability density function (PDF) and the model uncertainty can ensure that the model and the parameters are good enough.

Several methods are available in the literature to solve the inference problem, estimate parameters, and the associated uncertainty. Bard (1974) presents methods to obtain the variance of models from the frequentist approach, including Least Squares and Maximum Likelihood methods. The main drawback of Bard (1974) methods is the hypothesis imposed to obtain the variance of the parameters, including known distribution, generally Gaussian. On the other hand, the Bayesian approach estimates the joint probability distribution using all available information about the system without assumptions about the target distribution (GAMERMAN; LOPES, 2006).

The Bayesian approach to inference allows obtaining the posterior PDF of any set of parameters ( $g_{\theta}(\eta \mid D, I)$ ) using information of the observation data ( $D$ ) and any previous information about the system ( $I$ ). Therefore, using the Bayes Theorem, it is possible to write the following relationship between the earlier variables (GAMERMAN; LOPES, 2006):

$$g_{\theta}(\eta \mid D, I) \propto L(\eta \mid D)g_{\theta}(\eta \mid I)^1, \quad (2.3)$$

where  $\eta$  represents sampled values of  $\theta$ ,  $L$  is the likelihood function, and  $g_{\theta}(\eta \mid I)$  is the prior distribution of  $\theta$ , that is a new observation of  $\theta$ . Equation 2.3 allows updating the actual knowledge of the system represented by the posterior  $g_{\theta}(\eta \mid D, I)$ .

The likelihood  $L(\eta \mid D)$  is defined by Migon, Gamerman e Louzada (2014) as a function that associates the value of the probability  $g_{\theta}(\eta \mid I)$  with each  $\eta$  value. By defining the estimation process as a minor square problem, the objective will be to minimize a loss function that can be represented by a weighted least square estimator (WLSE). So, the likelihood function can be defined as:

$$L(\eta \mid D) \propto -\frac{1}{2} \sum_{i=1}^n (y_i^{exp} - y_i^m)^{\top} \Phi^{-1} (y_i^{exp} - y_i^m), \quad (2.4)$$

where  $(y_i^{exp} - y_i^m)$  is the residual between the experimental data ( $y_i^{exp}$ ) and the obtained with the prediction model ( $y_i^m$ ). Also, the use of the WLSE estimator implies the use of the variance of the residual between the  $y^{exp}$  and  $y^m$  for the  $ny$  outputs of the system, represented in Equation 2.4 as  $\Phi = diag[\Phi_1, \Phi_2, \dots, \Phi_{ny}]$ .

The posterior PDF of each  $\theta_i$  parameter of the vector  $\theta$ ,  $g_{\theta}(\eta \mid D, I)$  is obtained finding the marginal posterior density function  $g(\theta_1, \theta_2, \dots, \theta_{np})$  and is defined by Gamerman e Lopes (2006) as:

$$g_{\theta}(\eta \mid D, I) \propto \int_{\theta} (L(\eta \mid D)g_{\theta})(\eta \mid I)d\theta_{n-j} \quad (2.5)$$

Chapter 5 by Gamerman e Lopes (2006) presents several methods for solving the inference problem of Equation 2.5. The Markov Chain Monte Carlo methods have some

---

<sup>1</sup>The proportionality symbol is placed in this equation to guarantee theoretical coherence behind the inference process. Since the samples have not yet been taken, it cannot be assumed that the posterior density is equal to the product of the prior and likelihood, as samples remain. This discussion is carried out in Section 1.5.1.

interesting features among the numerical integration methods. Among these characteristics, convergence is more important because when chains are adequately constructed, after a sufficiently high number of iterations, the chains will converge to an equilibrium distribution. Thus, Figure 2.2 shows a schematic diagram of the solution to the inference problem using the MCMC method. The general idea is that the MCMC uses existing information, such as experimental data and a likelihood function, to provide a mathematical model and the associated PDF of the estimated parameters.

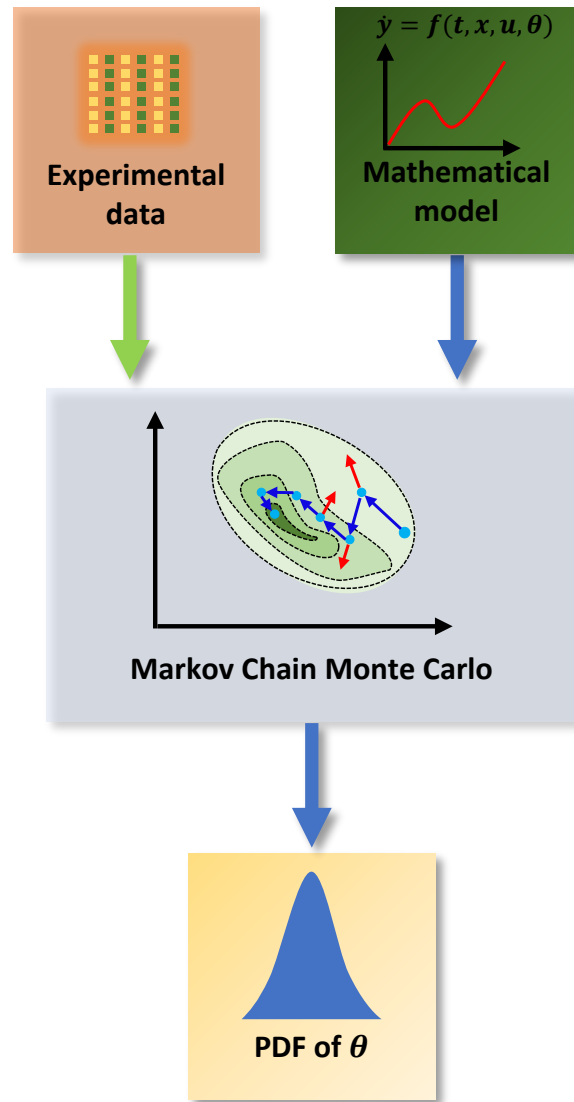


Figure 2.2: MCMC method.

On the other hand, this Chapter proposes using the DRAM (Delayed Rejection Adaptive Metropolis) MCMC algorithm that Haario et al. (2006b) presented to solve the inference problem. The DRAM algorithm combines the Adaptive Metropolis, which provides global adaptation, and the Delayed Rejection, which offers local adaptation. Also, the



main idea of the DRAM algorithm is to collect information during the chain run and tune the target PDF by using the learned information.

### 2.2.2 Synthetic data generation

In building AI models, data quality is crucial to obtain great-adjusted models. This Chapter proposes a methodology to build AI models using synthetic data from a non-linear model. Also, the methodology is based on the uncertainty propagation law established in BIPM et al. (2008, 2008, 2011). In this sense, the quantity and quality of data used in training must be adequate. Non-linear models in optimization and control applications often represent a high computational cost. These computational efforts can be reduced using artificial intelligence models trained with data from validated non-linear models.

This paper proposes to build the training database by drawing a sample of the parameters PDF and propagating it to the outputs of the non-linear model. In this way, the data used for training the models need to be representative of the operating conditions of the system and characterize the uncertainties of the non-linear model. Figure 2.3 presents a generic Monte Carlo Method (MCM) simulation scheme for synthetic data generation. Figure 2.3 scheme is inspired by the algorithm for implementing the Monte Carlo Method presented in Supplement 1 and 2 to the "Guide to the expression of uncertainty in measurement" (BIPM et al., 2008, 2011).

The first step of the algorithm is to determine the number  $m$  of trials to be performed. In general, the MCM will produce better responses with more shots, and the recommendation is to use a value  $m = 10^6$  (MIGON; GAMERMAN; LOUZADA, 2014). However, smaller values can be used when evaluating complex models requiring high computational costs for numerical solutions. This reduction in the number of tests may not allow the correct characterization of the PDF of the output values and produce less reliable results (BIPM et al., 2008).

After defining the number of tests to be performed, a matrix  $\Theta$  containing  $m$  vectors of PDF samples of  $g_\theta(\eta \mid D, I)$  is built:

$$\Theta = \begin{pmatrix} \theta_{1,1} & \dots & \theta_{1,np} \\ \vdots & \ddots & \vdots \\ \theta_{m,1} & \dots & \theta_{m,np} \end{pmatrix}. \quad (2.6)$$

Thus,  $m$  distinct set of model parameters are obtained so that it is possible to write

$$\dot{\mathbf{y}}_{1,\dots,m} = \mathbf{f}(t, \mathbf{y}, \mathbf{u}, \Theta), \quad (2.7)$$

This allows integration of the model so that the  $m$  dynamic responses are obtained for the output variables  $y$ .

As a dynamic model, obtaining representative data from the entire operating region is necessary. An independent pseudorandom binary Sequence (PRBS) signal for each of the  $u$  inputs is generated through a Latin Hypercube Sampler (LHS). The PRBS signals are then combined with the system's non-linear mathematical model. A dynamic response is received for each parameter combination obtained from the PDF of the parameters. Additionally, the system's dynamic response will be represented by a set of curves obtained

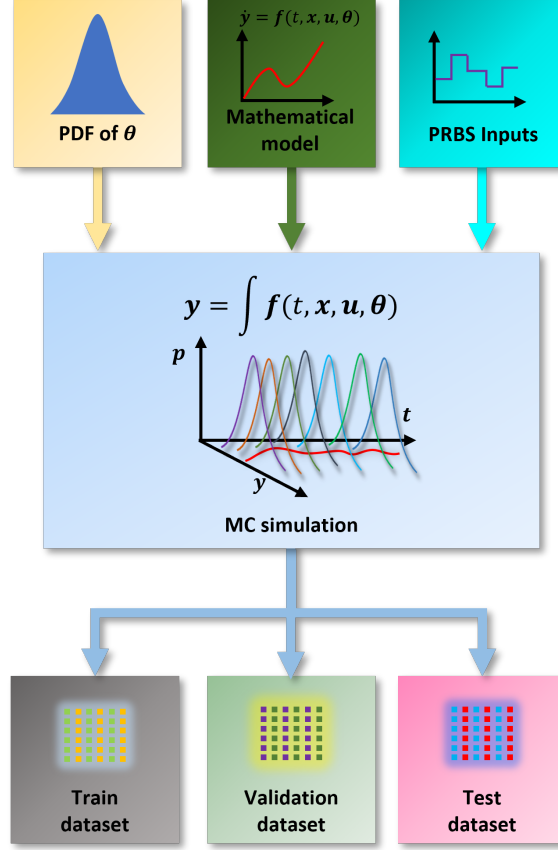


Figure 2.3: Monte Carlo simulation method to data generating.

from this Monte Carlo (MC) simulation. In this scenario of propagation of uncertainties by MC in dynamical systems, the actual value must be calculated for each sampling instant in which the equations were solved. In this way, it is considered that for each sampling instant, a PDF sample is obtained for each system output response  $y(t)$ .

From Figure 2.3, it is also possible to observe that the data generated through the MC simulation will be divided into three sets: training, testing, and validation. This division follows the standard literature guidelines for training AI models (HAYKIN, 1999). The first two sets are used during the supervised training of the models since the algorithms need two data sets for training and validation. The third set, the test dataset, is used for final cross-validation. Thus, the ability to predict new data is evaluated. Additionally, dividing the data into three sets must consider that the training process has consistent information. In this sense, sets are typically divided into Train - 70%, Validation - 15%, and Test - 15%.

### 2.2.3 Data curation

A database to train a dynamic data-driven model should be systematically organized to represent the system dynamics. Hence, the identified model can approximate the observed dynamic phenomena. There are several ways to manage a data set to incorporate the

system's dynamics. The most common is Non-linear autoregressive with exogenous inputs (NARX) predictors. The general NARX structure is composed of a prediction of the actual output as follows:

$$y_k = f(u_k, u_{(k-1)}, u_{(k-2)}, \dots, u_{(k-n)}, y_{(k-1)}, y_{(k-2)}, \dots, y_{(k-p)}) + \epsilon \quad (2.8)$$

where  $u_{(k-1)}, u_{(k-2)}, \dots, u_{(k-n)}$  are the  $n$  input delay, and  $y_{(k-1)}, y_{(k-2)}, \dots, y_{(k-p)}$  are the  $p$  of past values for the output. The noise,  $\epsilon$ , is additive: for the NARX, the error information is assumed to be filtered through the system's dynamic. Equation 2.8 shows that the NARX predictor has two hyperparameters:  $n$  and  $p$ . These parameters should be defined correctly to improve the dynamic representativeness of the data. For this purpose, He e Asada (1993) proposed the Lipschitz coefficient analysis. A Lipschitz coefficient is then calculated for each pair of measurements. For further information about the Lipschitz coefficients calculations, see He e Asada (1993) and in Section 1.5.2.

#### 2.2.4 Building the AI model

Figure 2.4 shows a methodology step to find hyperparameters of the network. That step defines the appropriate architecture one wants before starting the training. The type of neurons is the first aspect to be evaluated when building a neural network. In this way, expert knowledge must be considered to define whether a network will be used, e.g., recurrent, convolutions, or feedforward. Choosing the network depends on the characteristic of the system being modeled and what is the main application.

Once the general architecture of the network is defined, the next step is to define its architecture. There are some powerful algorithms available in the literature for this means. The optimal number of layers determines the architecture of a neural network, the number of neurons per layer, and activation functions, among others (LI et al., 2016). Determining these parameters is one source of uncertainties during the modeling process that needs to be considered.

#### 2.2.5 Monte Carlo training

The fourth step of the main methodology presented in Figure 2.1) is the uncertain training of networks through Monte Carlo simulations. The stages of generating data, obtaining the architecture of the neural network, and the hyperparameters provide the necessary information for the uncertainty training of the networks. In this way, it is sought to characterize the prediction region of the identified non-linear system, obtained in Sec. 2.2.1, through a set of networks capable of representing each of the probable outputs of the model.

Figure 2.5 presents a simplified schematic diagram of this step in the methodology. It is possible to observe that the MC Training stage boils down to massive training on top of the generated data. The result is a set of equally probable trained AI models that comprehensively account for the uncertainty sources.

This training step follows the concepts of Monte Carlo simulations and requires the most computational effort. The simulation is built to train a neural network with aleatory

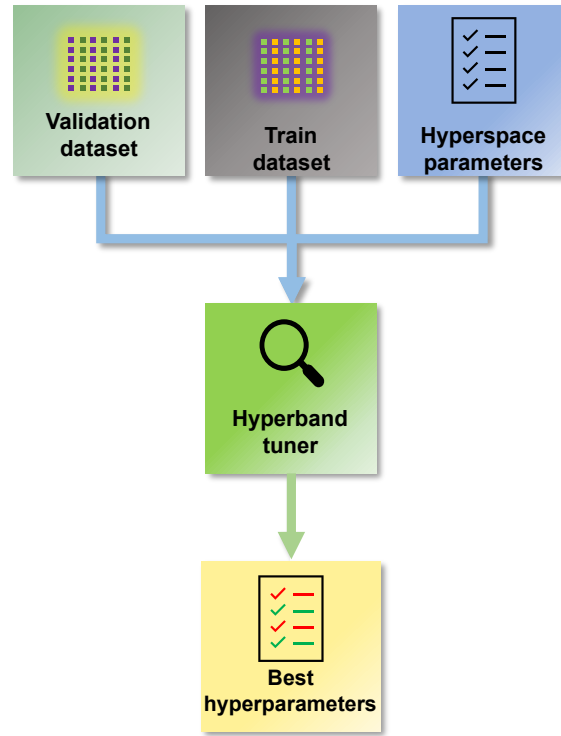


Figure 2.4: Optimization procedure to find the hyperparameters of the neural network.

initial weights and biases and optimal architecture for the training, validation, and test datasets. In addition to the computational effort required to perform this step, the volume of data used will also be significant.

On the other hand, nowadays, SciML model training has become increasingly efficient. The literature has presented algorithms that extract maximum performance from the available hardware. Additionally, manufacturers have built hardware with specific characteristics for application training and execution of artificial intelligence. In this scenario, the available technology makes running a Monte Carlo simulation for neural network training feasible.

### 2.2.6 Propagation and cross-validation

The methodology proposed in this Chapter includes a supplementary validation stage of the built AI model. In this step, the data used for cross-validation are labeled "Test data" in Figure 2.3 (Note that the set called validation is used during the training.). In a complementary way, the validation in the context of dynamic models with uncertainty needs to be evaluated to compare the coverage regions. In this context, a model is considered validated when the coverage of regions of models is overlapping, implying that the predictions are statistically equal.

The main issue involved is the method used to assess the uncertainty of the model. This chapter proposes using the Monte Carlo method for the non-linear phenomenological and the AI models. Thus, the uncertainty of the evaluated model can be obtained as-

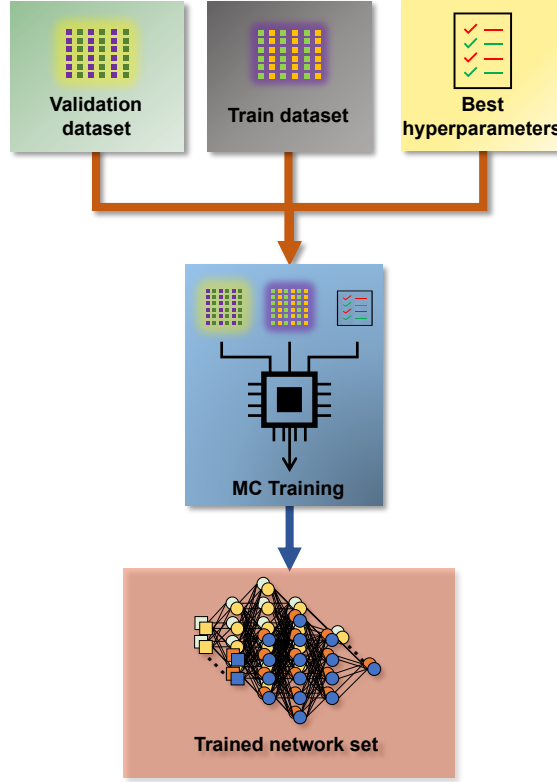


Figure 2.5: Monte Carlo training method.

suming the same hypotheses proposed by Haario, Saksman e Tamminen (2001a), Haario et al. (2006b). That is, an inverse Gamma distribution approximates the variance. Then:

$$V[y] \approx \Gamma^{-1}(x, \alpha, \beta) \quad (2.9)$$

where the distribution is supported in  $x > 0$  and represented by  $\Gamma^{-1}$  and the  $\alpha$  and  $\beta$  parameters are the shape and scale of the distribution.

In its turn, to obtain the parameters,  $\alpha$  and  $\beta$ , Gelman et al. (2013) suggest using:

$$\alpha(j) = \frac{N_{prior}(j) + N_{data}(j)}{2}, \quad (2.10)$$

$$\beta(j) = \frac{2}{N_{prior}(j) \cdot V_0^2 + SSE(j)}, \quad (2.11)$$

where  $j = 1, 2, \dots, ny$  is the number of outputs of the model and  $N_{prior}$  is the number of observations in the prior PDF and  $N_{data}$  is the number of experimental data. On the other hand,  $V_0^2$  is the variance of Prior, and  $SSE(j)$  is the sum of the squared errors between the prediction and the experimental data.

Using the hypothesis of non-informative prior, the prior variance is unknown, and the number of observations is zero. In this way, the previous equations can be approximated by:

$$\alpha(j) = \frac{N_{data}(j)}{2}, \quad (2.12)$$

$$\beta(j) = \frac{2}{SSE(j)}. \quad (2.13)$$

The methodology of this Chapter implies obtaining two sets of model parameters that will each have their associated uncertainty. Thus, the proposal is that the variance calculation is performed using the SSE obtained through the estimation data when the MCMC is performed. The SSE is obtained with the training data for network training. Then, the uncertainty of a prediction will be based on the variance of the model that will follow the inverse gamma distribution. This methodology allows for characterizing the epistemic uncertainty of the model.

## 2.3 RESULTS AND DISCUSSION

This section presents the results of applying the proposed methodology in a case study. A polymerization reactor with synthetic data was used as a case study. Sec. 2.3.1 presents the detailed polymerization model. Next, following the proposed methodology, it explains networks' construction using the Monte Carlo method, propagation, and final cross-validation.

### 2.3.1 Case study: Polymerization reactor

The reactor model is presented in detail by Alvarez e Odloak (2012). It is composed of a system of algebraic differential equations (DAE) with thirteen equations, as follows:

$$\frac{d[I]}{dt} = \frac{Q_i[i_f] - Q_t[I]}{V} - k_d[I], \quad (2.14)$$

$$\frac{d[M]}{dt} = \frac{Q_m[M_f] - Q_t[M]}{V} - k_p[M][P], \quad (2.15)$$

$$\frac{dT}{dt} = \frac{Q_i[T_f - T]}{V} + \frac{-\Delta H_r}{\rho C_p} k_p[M][P] - \frac{hA}{\rho C_p V} (T - T_c), \quad (2.16)$$

$$\frac{dT_c}{dt} = \frac{Q_c(T_{cf} - T_c)}{V_c} + \frac{hA}{\rho C_p V} (T - T_c), \quad (2.17)$$

$$\frac{dD_0}{dt} = 0.5k_t[P]^2 - \frac{Q_t D_0}{V}, \quad (2.18)$$

$$\frac{dD_1}{dt} = M_m k_p[M][P] - \frac{Q_t D_1}{V}, \quad (2.19)$$

$$\frac{dD_2}{dt} = 5M_m k_p[M][P] + M_m \frac{k_p^2}{k_t} [M]^2 - \frac{Q_t D_2}{V}, \quad (2.20)$$

$$[P] = \left[ \frac{2f_i k_d [I]}{k_t} \right]^{0.5}, \quad (2.21)$$

$$Q_t = Q_i + Q_s + Q_m, \quad (2.22)$$

$$\bar{M}_w = M_m \frac{D_2}{D_1}, \quad (2.23)$$

$$PD = M_m \frac{D_2 D_0}{D_1^2}, \quad (2.24)$$

$$\eta = 0.0012(\bar{M}_w)^{0.71}. \quad (2.25)$$

In the above DAE system, Equation 2.14 to 2.17 represent the mass and energy balance of the Monomer and Initiator. Equation 2.18 to 2.20 are the moments' equations of the dead polymer, in which  $D_0$ ,  $D_1$ , and  $D_2$  represent the moments of the dead polymer. Algebraic equations are used to describe the relationship between supplementary variables. Equation 2.23 represents the weight-average molecular weight, and Equation 2.25 represents the viscosity. Tab. 2.1, 2.2 and 2.3 show the model's parameters' general definition, the initial condition value of inputs and steady-state of outputs systems variables.

Alvarez e Odloak (2012) developed this model based on seven hypotheses which are: the lifetime of the radical polymer is shorter than other species; Long Chain Assumption (LCA) related to the monomer consumption; the chain transfer reaction to monomer and solvent can be neglected; operation below 373K because greater temperatures cause monomer thermal initiation; termination by disproportionation is not considered; the rate of termination is dominant; only the heat of polymerization is considered.

Alvarez e Odloak (2012) discuss the polymerization reactor from the control and optimization point of view. However, other relevant aspects are pointed out. One of

Table 2.1: Parameters and initial conditions.

Nominal Process Parameters	Value
Frequency factor for initiator decomposition, $A_d(h^{-1})$	$2.142 \times 10^{17}$
Activation energy for initiator decomposition, $E_d(K)$	14897
Frequency factor for propagation reaction, $A_p(L \cdot mol^{-1} \cdot h^{-1})$	$3.81 \times 10^{10}$
Activation temperature for propagation reaction, $E_p(K)$	3557
Frequency factor for termination reaction, $A_t(Lmol^{-1}h^{-1})$	$4.50 \times 10^{12}$
Activation temperature for termination reaction, $E_t(K)$	843
Initiator efficiency, $f_i$	0.6
Heat of polymerization, $-\Delta H_r(J \cdot mol^{-1})$	$6.99 \times 10^4$
Overall heat transfer coefficient, $hA(J \cdot K^{-1} \cdot L^{-1})$	$1.05 \times 10^6$
Mean heat capacity of reactor fluid, $\rho C_p(JK^{-1}L^{-1})$	1506
Heat capacity of cooling jacket fluid, $\rho_c C_{pc}(JK^{-1}L^{-1})$	4043
Molecular weight of the monomer, $M_m(g \cdot mol^{-1})$	104.14
Initial conditions	Value
Reactor volume, $V(L)$	3000
Volume of cooling jacket fluid, $V_c(L)$	3312.4
Concentration of initiator in feed, $I_f(mol \cdot L^{-1})$	0.5888
Concentration of monomer in feed, $M_f(mol \cdot L^{-1})$	8.6981
Temperature of reactor feed, $T_f(K)$	330
Inlet temperature of cooling jacket fluid, $T_{cf}(K)$	295

these, Alvarez e Odloak (2012), uses Equation 2.24 as a virtual analyzer for the viscosity because this is a complex variable to measure in the studied reactor. The authors use temperature and viscosity as controlled variables to manipulate the initiator flow rate and the rate of the cooling jacket. Therefore, this case study is used to validate the proposed methodology for uncertainty assessment of neural networks. Also, it is shown that using an AI model reduces the computational efforts of the control and optimization loops.

As no experimental data of the measured variables is available regarding this system, this paper proposes using random white noise to simulate the interferences that usually occur in an experimental setup. The system was simulated with the initial and steady-state conditions in Tab. 2.2 and 2.3.

Table 2.2: Steady-state inputs conditions and LHS region.

Variable	Steady-state	Minimum	Maximum
Flow rate of initiator, $Q_i(L \cdot h^{-1})$	108	91.8	124.2
Flow rate of solvent, $Q_s(L \cdot h^{-1})$	3312.4	2815.5	3809.26
Flow rate of monomer, $Q_m(L \cdot h^{-1})$	0.5888	0.5005	0.6771
Flow rate of cooling jacket fluid, $Q_c(L \cdot h^{-1})$	8.6981	7.3934	10.0028



Table 2.3: Output variables at steady-state.

Variable	Value
Concentration of initiator in the reactor, $I(\text{mol.L}^{-1})$	330
Concentration of monomer in the reactor, $I(\text{mol.L}^{-1})$	295
Temperature of the reactor, $T(K)$	323.56
Temperature of cooling jacket fluid, $T(K)$	305.17
Molar concentration of dead polymer chains, $D_0(\text{mol.L}^{-1})$	$2.7547 \times 10^{-4}$
Mass concentration of dead polymer chains, $D_1(\text{g.L}^{-1})$	16.110

Figure 2.6 shows the LHS generated with  $\pm 15\%$  of the steady-state input value and used as input in the simulation. A total of 30 steps, with 150 hours of simulation each, were developed to compose the synthetic data.

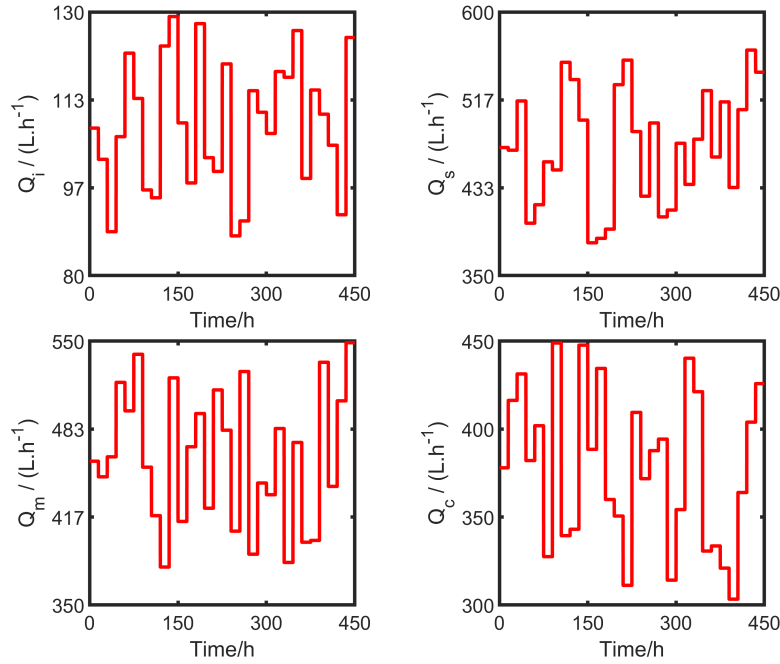


Figure 2.6: LHS inputs.

Figure 2.7 presents the correlation heat map for the variables set by the LHS. The main diagonal presents high correlations as it pairs the variables with themselves. In contrast, the pairing between the other entries produces a value that allows verifying if the generated values are uncorrelated. As low as these values are, they will be less correlated. As shown in Figure 2.7, the correlations are close to null, demonstrating that the LHS can efficiently generate uncorrelated samples.

The synthetic output dataset is shown in Figure 2.8. As mentioned, random noise with -20dB and 10% of the output range is included in the signal to emulate the field

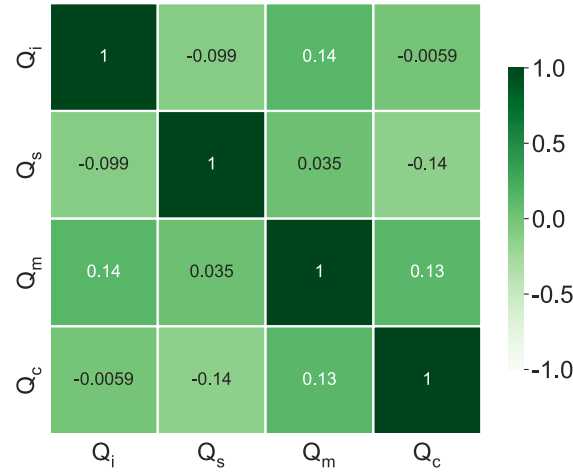


Figure 2.7: Inputs correlation map.

conditions. All these variables are used in the Likelihood function presented in Equation 2.4. 70% of this data is used for the SciML model identification, and the rest is used for the methodology cross-validation step, as discussed in the subsequent sections.

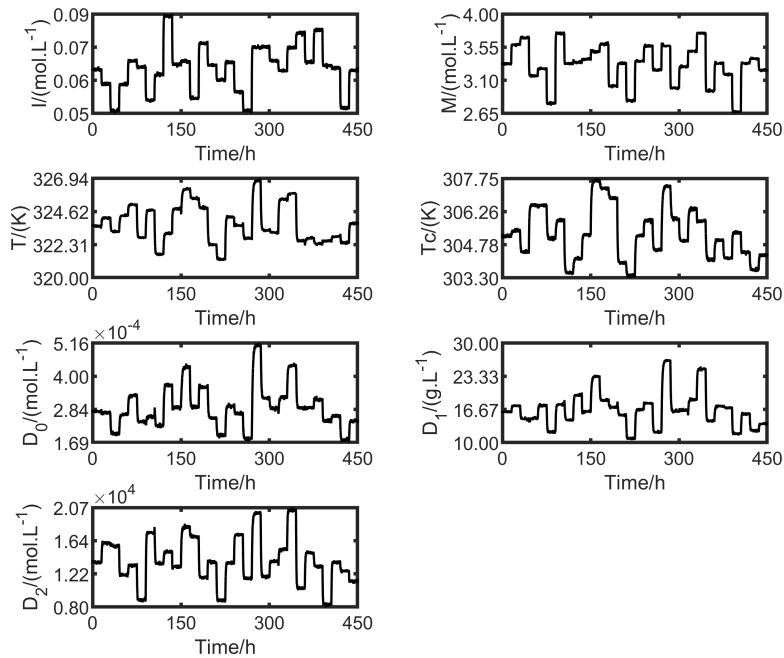


Figure 2.8: Synthetic output data with white noise.

### 2.3.2 MCMC

The generated synthetic data allows the uncertainty assessment of the Alvarez e Odloak (2012) model, done through the MCMC methodology. Therefore, the 12 model parameters and six initial conditions from Table 2.1 are used as decision variables for the MCMC algorithm. The DRAM algorithm proposed by Haario et al. (2006b) allows limiting the parameters search region. Therefore, in this paper, the search region was limited to  $\pm 5\%$  of the nominal value of the parameter. Also, all parameters were normalized to the nominal value to facilitate the algorithm's convergence.

Other algorithm aspects must be set. The first is the number of samples sorted to build the joint PDF. At this point, the algorithm was configured to create a set with 30,000 samples of the target joint PDF. The algorithm was also configured to use a non-informative prior. Therefore, the algorithm chooses 5000 samples to update the prior and discard the chain to estimate the parameter variance. After the burn step, the MCMC algorithm restarts to build the chains and evaluate the parameters in the search region.

Tab. 2.4 shows the resulting normalized parameters obtained from the Markov Chains. The MCMC algorithm does not make any assumptions about the target distribution. The mean and median in normal distributions converge to the same number. However, it is more conservative to assume that the distribution is not Gaussian and use the median as a value for the most probable value. In Tab. 2.4 it is possible to see the difference between the parameters mean and median. Tab. 2.4 also shows the standard deviation (std) and the geweke diagnose parameter (BROOKS; ROBERTS, 1997). For some parameters, the std value is relatively high. However, the geweke parameter indicates that the chain converges (BROOKS; ROBERTS, 1997). Figs. A.1 to A.11 in the supplementary material (Appendix A) shows the confidence region and the entire Markov Chain. Those figures compare Gaussian and unshaped regions proposed by Possolo (2010).

### 2.3.3 Synthetic data generation for training

The results of the MCMC allow the construction of the parameters PDF of the phenomenological model capable of representing the uncertainty of the non-linear system. Thus, randomly selecting a set of parameters is possible to build a set of non-linear responses. This work randomly selected a sample of 10000 distinct non-linear parameters. Subsequently, all resulting models were excited with the same LHS signal as in Figure 2.6, and the result was 10000 different dynamic responses.

With these dynamic trajectories, it was possible to establish the number of embedded dimensions of the NARX model. The Lipschitz method is presented in Sec. 2.2.3 and used to define the NARX regressors. Figure 2.9a and 2.9b show these results. In Figures 2.9a and 2.9b, the decisive factor is the slope of the surface because when there is a high variation between two delays, the increase is considered important. However, if the slope variation is low, it can be considered that this inclusion is not necessary. Then, it is possible to observe that a delay of four sampling instants for the inputs and one for the output variables is reasonable for a good representation, as after these values, the slope starts to be constant.

Table 2.4: Normalized parameters obtained by DRAM algorithm.

Parameter	Mean	Median	STD	Geweke
$A_d$	$1.00 \times 10^0$	$1.01 \times 10^0$	$1.96 \times 10^{-2}$	$9.95 \times 10^{-1}$
$E_d$	$1.00 \times 10^0$	$1.00 \times 10^0$	$6.97 \times 10^{-4}$	$9.99 \times 10^{-1}$
$A_p$	$9.99 \times 10^{-1}$	$9.98 \times 10^{-1}$	$1.99 \times 10^{-2}$	$9.96 \times 10^{-1}$
$E_p$	$9.98 \times 10^{-1}$	$9.98 \times 10^{-1}$	$3.17 \times 10^{-3}$	$9.99 \times 10^{-1}$
$A_t$	$1.00 \times 10^0$	$1.00 \times 10^0$	$2.06 \times 10^{-2}$	$9.83 \times 10^{-1}$
$E_t$	$9.91 \times 10^{-1}$	$9.88 \times 10^{-1}$	$2.42 \times 10^{-2}$	$9.91 \times 10^{-1}$
$f_i$	$1.00 \times 10^0$	$1.01 \times 10^0$	$2.33 \times 10^{-2}$	$9.94 \times 10^{-1}$
$-\Delta H_r$	$9.99 \times 10^{-1}$	$1.00 \times 10^0$	$7.93 \times 10^{-3}$	$9.96 \times 10^{-1}$
$hA$	$1.00 \times 10^0$	$1.00 \times 10^0$	$8.11 \times 10^{-3}$	$9.96 \times 10^{-1}$
$\rho C_p$	$9.97 \times 10^{-1}$	$9.98 \times 10^{-1}$	$9.80 \times 10^{-3}$	$9.96 \times 10^{-1}$
$\rho_c C_{pc}$	$1.01 \times 10^0$	$1.01 \times 10^0$	$9.63 \times 10^{-3}$	$9.97 \times 10^{-1}$
$M_m$	$9.99 \times 10^{-1}$	$9.99 \times 10^{-1}$	$6.41 \times 10^{-4}$	$9.99 \times 10^{-1}$
$V$	$9.98 \times 10^{-1}$	$9.99 \times 10^{-1}$	$3.71 \times 10^{-3}$	$9.99 \times 10^{-1}$
$Vc$	$1.00 \times 10^0$	$1.00 \times 10^0$	$1.34 \times 10^{-2}$	$9.99 \times 10^{-1}$
$If$	$1.00 \times 10^0$	$1.00 \times 10^0$	$3.15 \times 10^{-4}$	$9.99 \times 10^{-1}$
$Mf$	$9.99 \times 10^{-1}$	$9.99 \times 10^{-1}$	$2.79 \times 10^{-4}$	$9.99 \times 10^{-1}$
$Tf$	$9.99 \times 10^{-1}$	$9.99 \times 10^{-1}$	$1.83 \times 10^{-4}$	$9.99 \times 10^{-1}$
$Tcf$	$1.00 \times 10^0$	$1.00 \times 10^0$	$4.32 \times 10^{-4}$	$9.99 \times 10^{-1}$

(a)

(b)

Figure 2.9: a) Lipschitz surface for reactor temperature; b) Lipschitz surface for polymer viscosity.

### 2.3.4 Monte Carlo training

The proposed methodology includes a step called Monte Carlo Training, composed of smaller steps. It starts with the identification of the Hyperparameters. It is followed by defining the data needed for training. Finally, the final Monte Carlo Training is done.

**2.3.4.1 Monte Carlo training** The network’s optimal structure is found by a search in a hyperspace formed by the structural parameters that constitute the networks. Tab. 2.5 shows the initial configuration included in the Hyperband algorithm (LI et al., 2016). In this way, the algorithm evaluated the network’s architecture, which ranged from two to six dense layers; each layer could have between 30 and 160 neurons. Other structural parameters were found in Tab. 2.5.

The results obtained from the hyperparameters search are presented in Tab. 2.6. It is possible to observe that a relatively more straightforward network was necessary to represent the temperature compared with the viscosity. This simpler architecture implies a big difference in the computational cost. The viscosity network required 6.4 more

Table 2.5: Hyperband hyperspace search.

Parameter	Search Space
Type of layer	Dense
Number of layers	2 – 6
Output layer	1
Activation function	<i>Relu</i> or <i>Tanh</i>
Number of neurons per layer	[30, 50, 70, 90, 100, 120, 130, 160]
Learning rate	0.0001, 0.001, and 0.1
Metrics	MAE – Mean Absolute Error
Loss	MSE – Mean Square Error

times to be trained than the network for the temperature. Also, networks have the same activation functions in the layers and the same learning rate. Tab. 2.6 also shows the MAE and MSE values resulting from the Hyperband search. The hyperband algorithm uses the training and validation datasets during the training. Then, the Test dataset is used after the training to test the final model parameters. It is noteworthy, however, that this methodology assumes that the network’s architecture is fixed; in other words, it assumes that the neural network architecture is good enough to represent the variable. In this way, it is considered that it is only necessary to execute the hyperparameter search process once. With the defined network structure, it is trained for all trajectories obtained from each non-linear model. Hence, a set of networks is identified, as described in Sec. 2.2.5.

Table 2.6: Resulting networks hyperparameters.

Hyperparameters	$T$	$\eta$
Number of layers	3	7
Neurons in each dense layers	[100, 90, 1]	[150, 90, 150, 90, 150, 90, 1]
Activation function	[ <i>tanh</i> , <i>tanh</i> ]	[ <i>tanh</i> , <i>tanh</i> , <i>tanh</i> , <i>tanh</i> , <i>tanh</i> , <i>tanh</i> ]
Initial learning rate	$1 \times 10^{-3}$	$1 \times 10^{-3}$
Trainable parameters	11081	71011
MSE Test	$2.37 \times 10^{-5}$	$5.84 \times 10^{-4}$
MAE Test	$4.30 \times 10^{-3}$	$2.03 \times 10^{-2}$

**2.3.4.2 Data size** An important aspect to be analyzed in neural network training is the guarantee of adequate training. In this sense, assessing whether the amount of information added to the training model is sufficient for the training algorithm to obtain a suitable model is necessary. Figure 2.10 the impact of the data in the neural network training of the polymer viscosity and reactor temperature. This evaluation was based on independent training sessions, in which the training started with aleatory weights

and biases, trained for a maximum of 300 epochs, and repeated twenty-five times. The first twenty-five training sessions were carried out using 100 experiments, each of which is a row of the training matrix and is considered an experiment; in other words, an experiment is a matrix that contains the regressors associated with the respective output. The average value of MAE and the final MSE were calculated. For the next twenty-five, 100 experiments were added, and so on. In Figure 2.10, it is possible to observe that for viscosity, after about 1750 experiments, there is no significant change in the MSE; however, for MAE, this value is about 1500. On the other hand, when the reactor temperature is evaluated, this value is higher, and more than 2500 experiments are needed for convergence. Thus, to ensure convergence, 3100 were used for both networks.

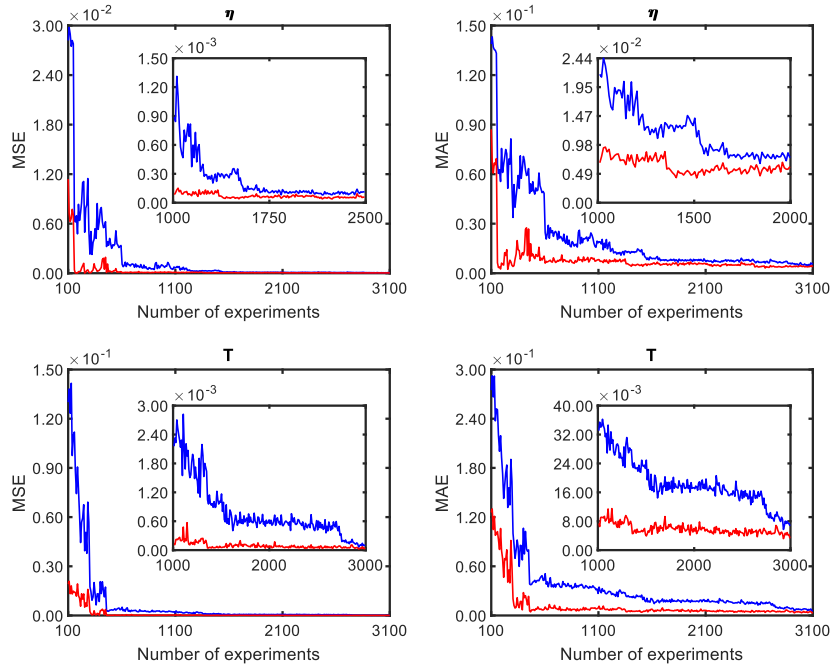


Figure 2.10: Training performance as a function of experiments. (—) Validation data; (—) Training data.

**2.3.4.3 Training** The adaptive moment estimation algorithm (ADAM) proposed by (KINGMA; BA, 2014) was used to train the chosen structure. The ADAM is an optimization method based on the descending gradient technique, making the ADAM algorithm efficient for problems involving extensive data and parameters. It also requires less memory than other training algorithms because the data is sliced into several packages and treated.

Building the networks involves an exhaustive training process called Monte Carlo training. The proposed methodology is based on the Monte Carlo method for PDF propagation. Thus, the assumed hypothesis is that the different trajectories generated through the non-linear model can represent the model's uncertainty. Therefore, training

networks capable of representing these distinct trajectories imply obtaining a PDF of trained parameters of an AI model, which also means the uncertainty of the model.

Convergence analysis of training networks via MC training can be performed by analyzing MAE and MSE values like conventional training. However, given the number of trained networks, it is more convenient to evaluate in histogram format as in Figure 2.11. In the first analysis, Fig 2.11 shows the histograms of the MAE and MSE indicators, both for the test and validation data. It is possible to observe that the histograms do not follow a Gaussian distribution. So, the mean may not be a good reference in statistical terms. Thus, Tab 2.7 shows the minimum, maximum, median, and standard deviation of the MAE and MSE.

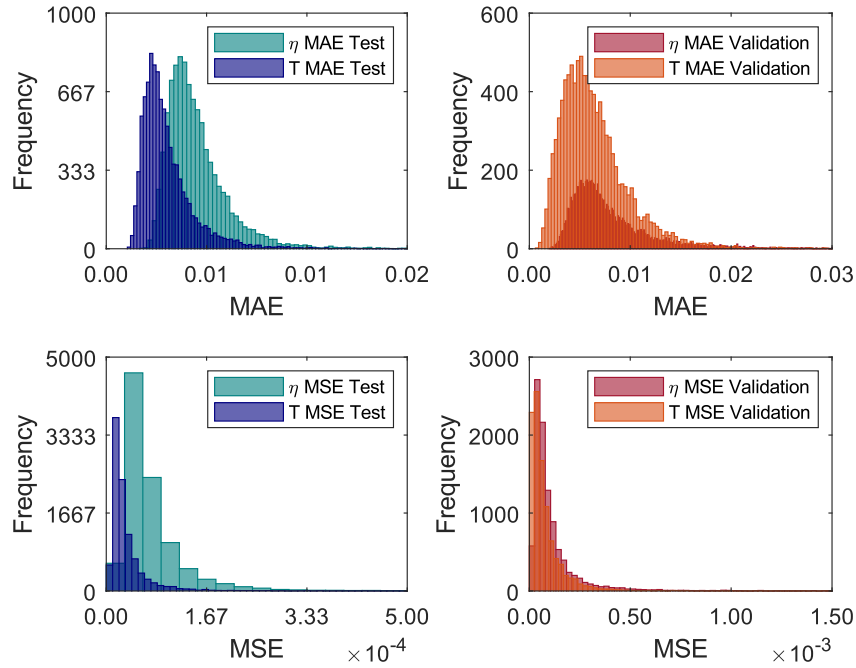


Figure 2.11: Training performance as a function of experiments.

In the Monte Carlo Training process, an Early Stopping option in MC training was used to reduce the risk of overfitting. Thus, training is aborted if no loss value is decreased for 100 epochs for the training and validation datasets. Figure 2.12 shows the histograms of the epochs trained during MC training for the two modeled variables. In the graph of Figure 2.12, the minimum number of epochs was 50, and the maximum was 300. Log-normal distributions should best represent this data type with lower or upper bounds because a limit implies the distribution is not symmetric. Figure 2.12 also shows that the data do not fit a lognormal distribution. Hence, no assumptions are made about the type of distribution of variables.

Table 2.7: Resulting networks hyperparameters.

	MAE Test	MAE Valid	MSE Test	MSE Valid
$T$				
Min	$1.49 \times 10^{-3}$	$7.91 \times 10^{-4}$	$5.64 \times 10^{-6}$	$2.73 \times 10^{-6}$
Max	$2.26 \times 10^{-3}$	$6.30 \times 10^{-2}$	$1.05 \times 10^{-3}$	$4.82 \times 10^{-3}$
Median	$3.57 \times 10^{-3}$	$5.77 \times 10^{-3}$	$2.37 \times 10^{-5}$	$4.98 \times 10^{-5}$
STD	$1.82 \times 10^{-3}$	$3.72 \times 10^{-3}$	$4.37 \times 10^{-5}$	$1.24 \times 10^{-4}$
$\eta$				
Min	$2.35 \times 10^{-3}$	$2.06 \times 10^{-3}$	$1.20 \times 10^{-5}$	$1.06 \times 10^{-5}$
Max	$7.35 \times 10^{-2}$	$2.01 \times 10^{-1}$	$1.82 \times 10^{-2}$	$5.27 \times 10^{-2}$
Median	$5.55 \times 10^{-3}$	$6.79 \times 10^{-3}$	$5.87 \times 10^{-5}$	$7.22 \times 10^{-5}$
STD	$3.38 \times 10^{-3}$	$6.23 \times 10^{-3}$	$4.55 \times 10^{-4}$	$9.25 \times 10^{-4}$

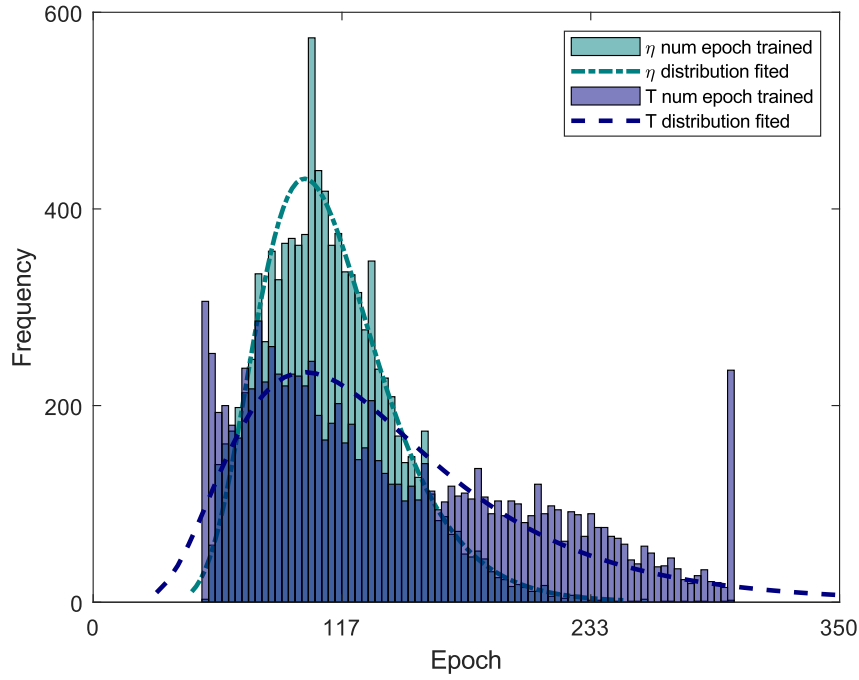


Figure 2.12: Trained epochs of Monte Carlo Training.

### 2.3.5 Uncertainty propagation and validation

The last step of the proposed methodology is the propagation of uncertainty and methodology cross-validation. The proposal is based on constructing the uncertainty regions of the neural network's prediction and comparing them with the uncertainty regions of the non-linear phenomenological model. With the prediction values, it is possible to calculate the variance of the networks using the same assumptions used to calculate the



uncertainty of the phenomenological model Equation 2.14 to 2.25. Figure 2.13 to 2.14 shows the comparisons between the predictions for the two modeled variables,  $T$ , and  $\eta$ .

Figure 2.13 to 2.14 presents the entire output dataset used for training, testing, and validation. A zoom is given to verify the variables' behavior in each region. It is pointed out, however, that the training and validation data sets are used during the training of the networks. In this way, it is unknown for the networks only the test data set used to certify their performance.

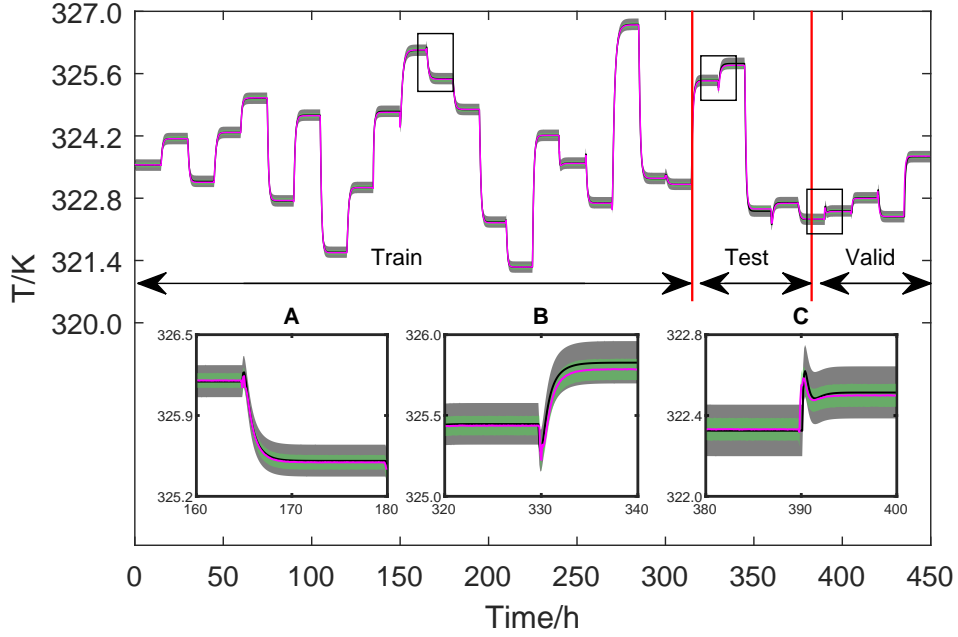


Figure 2.13: Training and validation data of  $T$ . (—) Nonlinear model; (—) NN model, (■) Nonlinear model uncertainty, (■) NN model uncertainty.

From a statistical point of view, when two measurements have overlapping coverage regions, it is impossible to differentiate, and they are considered equal. Therefore, the validation of the AI model against the non-linear phenomenological model is achieved when the dynamic range regions are superimposed. In this sense, the Graphs of Figure 2.13 to 2.14 allow us to conclude that both models statistically produce the same dynamic response.

## 2.4 CONCLUSION

This work presented a novel methodology for evaluating the uncertainty of Scientific Machine machine-learning models. A proposed approach considers several uncertainties associated with the SciML model structure, the data used, and the original data source. The proposed strategy comprised five steps: using the Markov Chain Monte Carlo method to obtain the uncertainty of the non-linear model parameters; generating synthetic data; identifying the neural network structure; training the Monte Carlo simulation; and validating the methodology and assessing the trained model.

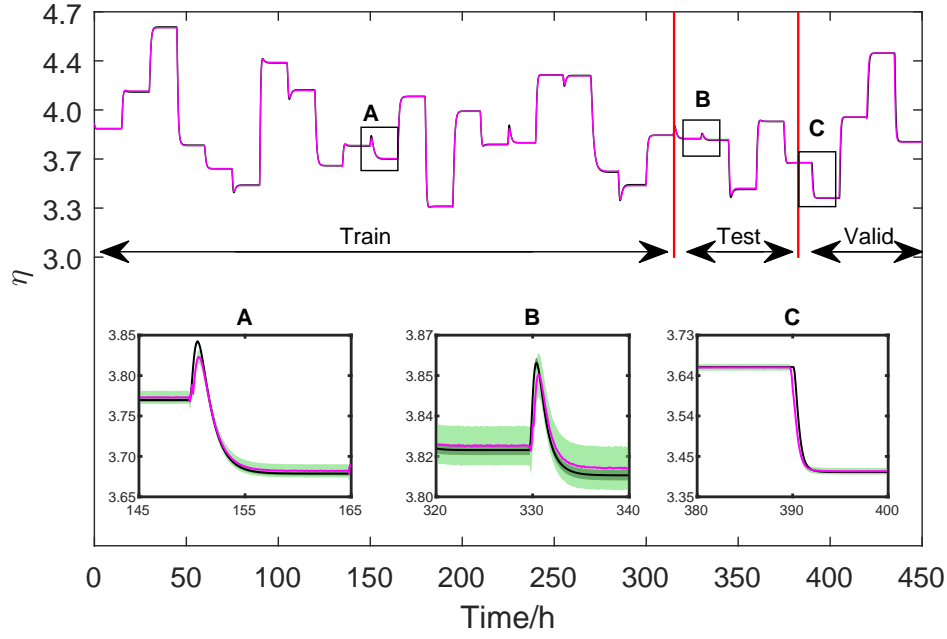


Figure 2.14: Training and validation data of  $\eta$ . (—) Nonlinear model; (—) NN model, (■) Nonlinear model uncertainty, (■) NN model uncertainty.

The proposed method considers epistemic and aleatory uncertainties. These uncertainties are considered in the context of the data used to train the models and the model itself. Therefore, providing an overall strategy for the uncertainty-aware models in the SciML field is possible. A case study demonstrated the method's consistency. Hence, two soft sensors were identified to provide information about the temperature and viscosity of a polymerization reactor. The results indicated that the soft sensor predictions are statistically equal to the validation data in dynamic and stationary regimes.

## CHAPTER 3

# A BAYESIAN APPROACH TO THE DYNAMIC MODELING OF ESP-LIFTED OIL WELL SYSTEMS: AN EXPERIMENTAL VALIDATION ON AN ESP PROTOTYPE.

Published on: **Costa, E. A.**; Abreu, O. S. L. de; Silva, T. d. O.; Ribeiro, M. P.; Schnitman, L. (2021). A Bayesian approach to the dynamic modeling of ESP-lifted oil well systems: An experimental validation on an ESP prototype. *Journal of Petroleum Science and Engineering*, 205, 108880.

URL: <https://doi.org/10.1016/j.petrol.2021.108880>.

### 3.1 INTRODUCTION

This chapter presents an integrated method for estimating parameters for an electric submersible pump system with process variables data. This Chapter addresses hypotheses 1 and 3, presented in Section 1.1, for constructing and validating non-linear models. A phenomenological model is validated as a case study to test the methodology. The parameters and the associated probability density function are obtained through Bayesian inference, and the model validation is achieved in two stages. The first one is validating the dynamic response in which the model is compared with the experimental data. The second is accomplished by comparing the regions covered by the experimental data and the steady-state model. The uncertainty of the experimental data is assessed using GUM (BIPM et al., 2008). In turn, the uncertainty of the model's prediction is obtained by propagating the probability density function parameters. The results indicate that the method can provide a model representing the system behavior within the existing uncertainties. Additionally, the procedure can be applied in oil production fields to provide substitute models for general purposes, such as production control, optimization, and assistance.

The remainder of this Chapter is organized as follows. Section 3.2 presents the validation method and includes the Bayesian formalism to obtain the joint PDF of the parameters and the GUM method's hypothesis for uncertainty assessment. Section 3.3 shows the experimental apparatus (3.3.1), the suitable dynamic model, and the validation and test results. The conclusions are offered in Section 3.4.

### 3.2 METHODOLOGY

The method proposed in this chapter aims to treat the ESP system parameters and obtain and assess uncertainty during the model validation systematic process. The proposed

technique is based on the traditional literature for parameter estimation and validation of dynamic models in uncertainty. The primary basis is the methods presented in Bard (1974), in which he discusses the fundamentals for estimating parameters of steady-state and dynamic models. From his proposition, obtaining the covariance matrices of different multivariable and nonlinear models is possible. Bard (1974) proposes calculating the parameters' covariance matrix by parameter perturbations. For dynamics and nonlinear models, convergence is challenging when finding the covariance matrix. On the other hand, the Bayesian inference provides a means of obtaining a sample of the model's estimated parameters' joint probability distribution. The Bayesian method does not have constraints if the model is dynamic or nonlinear.

The method proposed in this Chapter is described in Fig. 3.1 and divided into five stages. The insertion of existing information (a priori) into the parameter estimation process is performed in the first step, described as inputs. The data set used in the proposed method is divided into two subsets. The largest one is used for parameter estimation and direct validation (Experimental Data 1). The minor, in turn, is used for test purposes (Experimental Data 2). The motivation to use different data series derives from the need to provide more information during the parameter estimation process. Specifically, information about stationary and dynamic states is essential to obtaining the parameter values in estimating dynamic systems parameters.

The second step consists of obtaining the PDF of the parameters through Bayesian inference. The method used in this step is presented by Gamerman e Lopes (2006) (Chapter 3). The data and the model are constraints to the inference process, so they must be respected. The inference process solves the dynamic model in each iteration. The inputs collected in the prototype are also used in the simulation. The Markov chain Monte Carlo method solves the inference problem and obtains the joint PDF. Haario et al. (2006a) algorithms are used to solve the Markov chains.

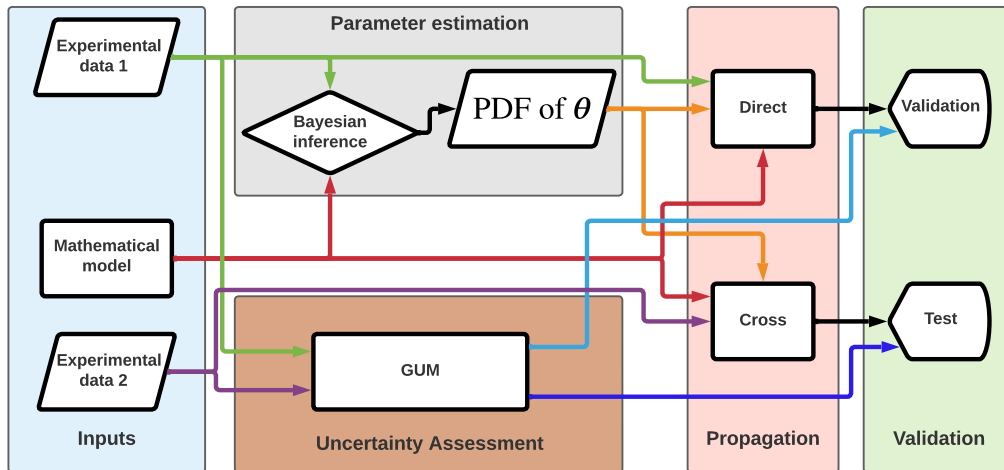


Figure 3.1: Proposed method for the validation of ESP models.

Parallel to obtaining PDFs, it is necessary to evaluate the uncertainty of the experi-

mental data. This step aims to reach the experimental data's confidence bound through the GUM by BIPM et al. (2008) method and its associated hypotheses.

The propagation and validation steps have two parallel and complementary ramifications. First, it is necessary to analyze the prediction model's behavior compared with the previous step's experimental data. This analysis aims to verify whether the parameters found can track experimental data or have significant deviations. In contrast, the second allows observation of the model's behavior concerning data different from that used to calculate the parameters. This latter analysis provides a better understanding of the ability to predict when using the model. The model will be validated when the prediction behavior has the desired characteristics. In this case, the parameters found become references for the prediction model. Specifically, the regions covered by the experimental data and the prediction model are compared in the validation stage. The model will be validated if there is an overlap of the confidence bounds.

### 3.2.1 Bayesian method for parameter estimation

Parameter estimation is prominent in the diagram of Fig. 3.1. The ESP dynamics models have a set of ' $n$ ' parameters to describe the behavior of the leading equipment of the system, generally,  $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$ . The numeric values of  $\boldsymbol{\theta}$  can be obtained using the factory information or parameter estimation process. The drawback of using the factory parameters is to deal with changing operating conditions or fluid properties. In contrast, using estimated parameters is more suitable to update them when necessary.

The literature reports several methods of parameter estimation. Bayesian inference is a method to obtain the parameter values and the associated PDF. This PDF is essential to characterize the parameters' uncertainty and, consequently, the uncertainty of the values predicted by the model. Furthermore, this method can handle nonlinear and multivariate models and appropriately addresses the current problem.

The Bayesian method consists of obtaining the posterior PDF of the parameters ( $\mathbf{g}_{\boldsymbol{\theta}}$ ), which is a function of the experimental data ( $D$ ) and the previous information about the system ( $I$ ). The relationships between these variables are written by Gamerman e Lopes (2006) as

$$\mathbf{g}_{\boldsymbol{\theta}}(\eta|D,I) \propto L(\eta|D)\mathbf{g}_{\boldsymbol{\theta}}(\eta|I), \quad (3.1)$$

where  $\eta$  are aleatory values of  $\boldsymbol{\theta}$ ,  $L$  is the likelihood and represents the behavior of residuals between the experimental data ( $y^{exp}$ ) and the predicted values obtained by the model ( $y^m$ ), and  $\mathbf{g}_{\boldsymbol{\theta}}(\eta|I)$  contains the probability distribution of  $\boldsymbol{\theta}$  before the observation of  $\eta$  and is called the prior density. Equation (3.1) means that the posterior  $\mathbf{g}_{\boldsymbol{\theta}}(\eta|D,I)$  is obtained by the combination between the likelihood  $L(\eta|D)$  and the prior  $\mathbf{g}_{\boldsymbol{\theta}}(\eta|I)$ .

The likelihood function can be defined using several functions. One of these is the weighted sum of square function as follows:

$$L(\eta|D) \propto -\frac{1}{2} \sum_{i=1}^n (y_i^{exp} - y_i^m)^{\top} \boldsymbol{\Phi}^{-1} (y_i^{exp} - y_i^m), \quad (3.2)$$

where  $\Phi = \text{diag}[\Phi_1, \Phi_2, \dots, \Phi_{ny}]$  are the variance of the residue between  $y^{exp}$  and  $y^m$ . Moreover,  $ny$  is the number of system outputs. In the form of Equation (3.2),  $L(\eta|D)$  is dimensionless. The marginal posterior density of  $\theta_j$  can be obtained from the joint density  $\mathbf{g}(\theta_1, \theta_2, \dots, \theta_{np})$  according to (GAMERMAN; LOPES, 2006):

$$\mathbf{g}_\theta(\eta|D, I) \propto \int_{np-1} L(\eta|D) \mathbf{g}_\theta(\eta|I) d\theta_{np-j}. \quad (3.3)$$

The optimal value of  $\theta$  will be defined by the expected value of  $\mathbf{g}_\theta(\eta|D, I)$ , in symbols

$$\hat{\theta} = \int_{-\infty}^{\infty} \eta \mathbf{g}_\theta(\eta|D, I) d\theta, \quad (3.4)$$

and the associated covariance matrix

$$U_{\theta\theta} = \int_{-\infty}^{\infty} (\eta - \hat{\theta})^\top (\eta - \hat{\theta}) \mathbf{g}_\theta(\eta|D, I) d\theta. \quad (3.5)$$

Gamerman e Lopes (2006) point out that the solution for multiple integrations cannot always be found analytically. Therefore, in these cases, the integrations can be solved using several methods, as i) approximations, ii) Monte Carlo, and iii) resampling. Among these methods, the Markov chain Monte Carlo (MCMC) seems to be appropriate to solve the inference problem of dynamic models, as suggested in (GAMERMAN; LOPES, 2006). Therefore, for the solution of Equation (3.4) and Equation (3.5) integrals', the method used is the MCMC solved by the adaptive metropolis algorithm proposed by Haario et al. (2006a).

### 3.2.2 Evaluation of the uncertainty of the experimental data

The uncertainty assessment must follow the validation process of models with experimental data. In this article, the validation process is supplemented by the uncertainty analysis of experimental data and parameters. Regarding the experimental data, uncertainty evaluation follows the GUM on BIPM et al. (2008) and the general concepts of the International Vocabulary of Metrology (VIM) on BIPM et al. (2012).

The GUM provides subsidies for assessing the uncertainty of experimental data. The evaluation of measurement uncertainty is done through the Guide's two components: Type A and Type B uncertainty. The standard type A uncertainty comes from the measurement's randomness, and its evaluation is indicated through statistical analysis. In turn, the standard type B uncertainty comes from existing information about the measurement process.

The type B uncertainty component contains all available information about the measurement system and includes accuracy, precision, resolution, and instrumental drift. The best source for obtaining information on standard B-type uncertainty is calibration certificates. These certificates are not always available. In these cases, manufacturer information or bench verification reports can be useful. However, it is considered that the less information about the measurement system, the greater the component of type B uncertainty (BIPM et al., 2008).

The final measurement result is written according to the most likely value (average or median) and the combined standard uncertainty associated with that measurement. The combined standard uncertainty is equal to the positive square root of a sum of Type A and Type B standard uncertainty terms (BIPM et al., 2008), in symbols:

$$u_c^2 = \sqrt{u_a^2 + u_b^2}. \quad (3.6)$$

The analysis provided by GUM has some inherent assumptions, including non-autocorrelated variables at steady state. They are separately treated.

**3.2.2.1 Steady-state** The steady-state hypothesis is valid when the experiments are maintained for long periods under the same input conditions. Therefore, in obtaining the experimental data, the inputs were kept fixed for periods equivalent to three times the system dynamics' settling time. Additionally, it is necessary to define the regions in which the dynamic modes are no longer active. This analysis is commonly performed visually for each system's output variables for which measurements exist. The appropriate graphical analysis of the steady states is described in Subsections 3.3.6 and 3.3.7.

**3.2.2.2 Autocorrelation** In continuous and dynamic processes, such as the ESP system, the hypothesis of non-autocorrelated variables is challenging to assume. However, through the adequate choice of sampling time, it is possible to minimize the dependence between the samples, allowing such a hypothesis to be assumed.

The autocorrelation of one of the process variables is shown in Fig. 3.2 and 3.3. It is possible to observe that with the initial sampling time of 1 second (Fig. 3.2), the samples' dependence exceeds two standard deviations, which corresponds to 95% confidence bounds. However, after resampling, the autocorrelation values are within the confidence limit (Fig. 3.3). Then, 100 seconds is the sampling time for the entire procedure presented in this paper.

The resampling procedure is performed only in parameter estimation and validation. Because the method used to evaluate the uncertainties of the experimental data (GUM) assumes non-autocorrelated variables. In solving the inference problem, the Bayesian approach does not have such constraints. However, if the parameters depend on the sampling time, the values obtained cannot be extrapolated to other sampling times. As shown in Subsection 3.3.2, the used parameters are not the system's sampling time-dependent. This independence makes the model suitable for control, using an appropriate sampling time.

### 3.3 RESULTS AND DISCUSSIONS

#### 3.3.1 Experimental apparatus

The schematic of the ESP apparatus used in this Chapter is presented in Fig. 3.4. The installation consists of a vertical well simulator with a height of 32 m. The pump was installed at a depth of 22.8 m from the top of the pipe. The production fluid is lubricant

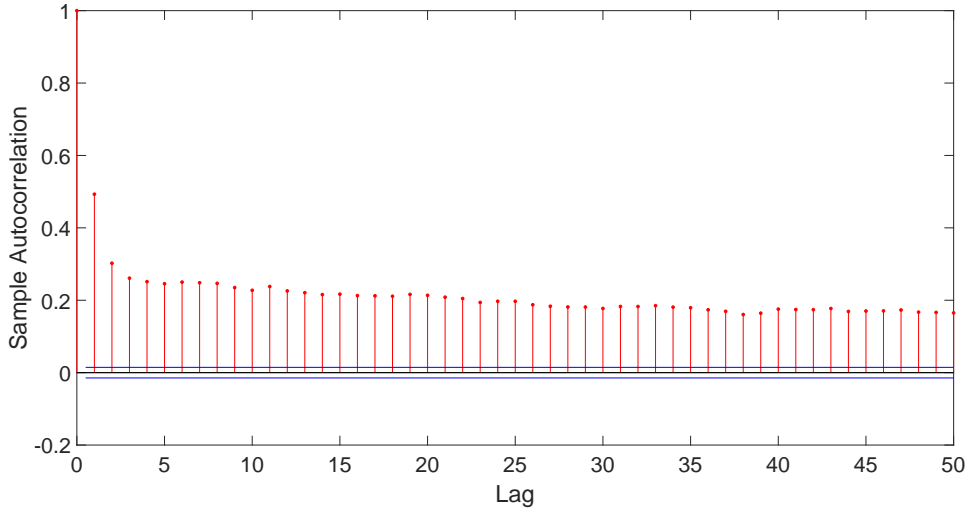


Figure 3.2: Autocorrelation of production flow with a sampling time of 1 second.

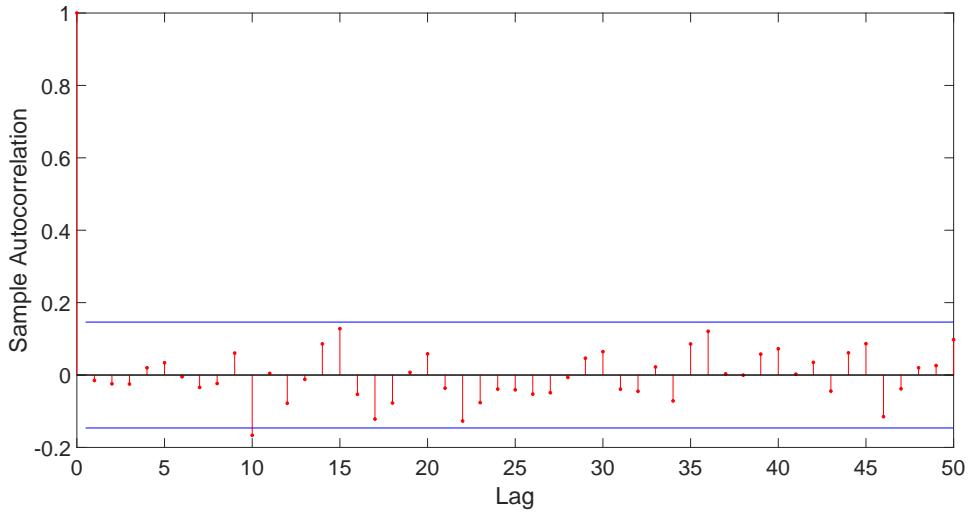


Figure 3.3: Autocorrelation of flow with a sampling time of 100 seconds.

oil with a mineral basis equivalent to oil with API 10 and viscosity equal to 0.00888667 *Pa.s*.

The installed pump has 15 stages and a nominal power of 18hp. The ESP system consists of a down-hole sensor, motor, seal, gas separator, and pump. Auxiliary systems are also used, such as a surface panel with a VSD (variable speed drive), compressors, circulation pumps, choke, and service valves.

The notation used in Fig. 3.4 and the additional deductions in this article are as follows:  $p_r$ ,  $p_{bh}$ ,  $p_{in}$ ,  $p_{wh}$  and  $p_m$  are pressures in the reservoir, the bottom of the well, intake pump, wellhead (choke pressure), and manifold, respectively. The volumetric flow rates are as follows:  $q_r$  is the flow rate into the well from the reservoir,  $q_m$  is the average



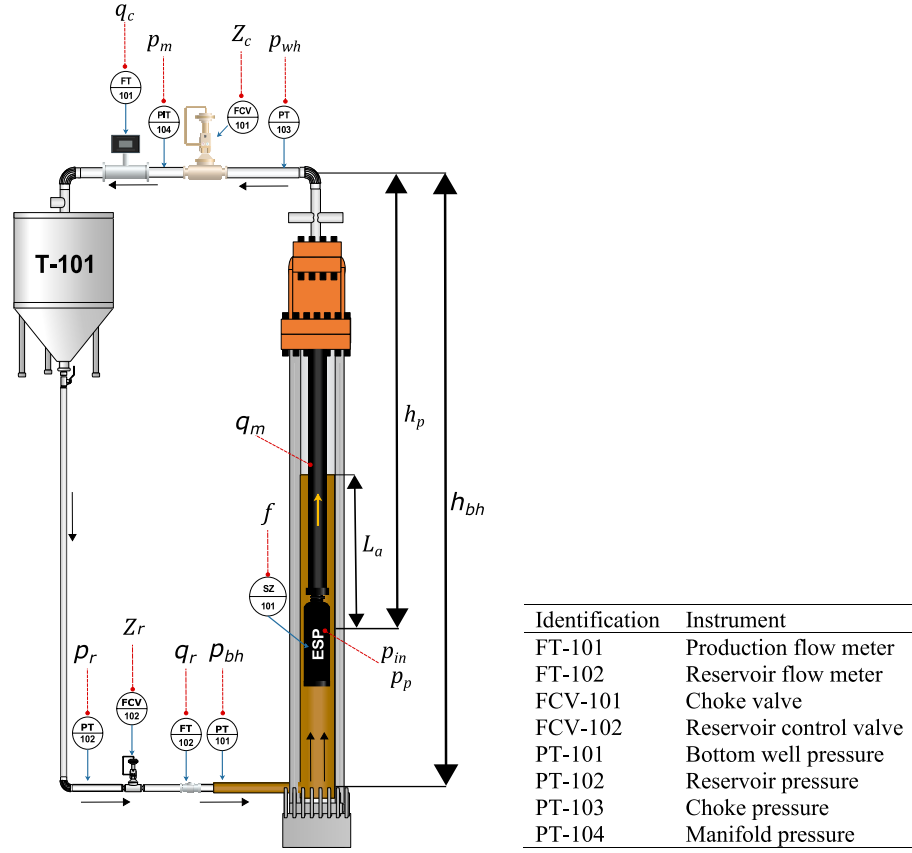


Figure 3.4: ESP-lifted well set up.

flow in the well, and  $q_c$  is the production choke flow rate. The valve refers to the height ( $h_c = 0m$ ), so the distance to the pump intake is  $h_p$ , and the distance to the bottom of the well is  $h_{bh}$ . Additionally, the liquid level in the annulus of the well ( $L_a$ ) is taken as a reference to the pump's intake, and  $f$  and  $Z_c$  are the system's inputs and are the ESP motor frequency and the choke valve opening, respectively.

### 3.3.2 Modeling

A simple model that has a good representation of the dynamic system in Fig. 3.4 can be represented by

$$\frac{dL_a}{dt} = \frac{1}{A}(q_r - q_m) = \frac{1}{A}(K_r Z_r \sqrt{(p_r - p_{bh})} - q_m), \quad (3.7)$$

$$\frac{dp_{wh}}{dt} = \frac{\beta_2}{V_2}(q_m - q_c) = \frac{\beta_2}{V_2}(q_m - K_c Z_c \sqrt{(p_{wh} - p_m)}), \quad (3.8)$$

$$\frac{dq_m}{dt} = \frac{\bar{A}}{\bar{\rho} \bar{l}}(p_{bh} - p_{wh} + \Delta p_p - \Delta p_h - \Delta p_f). \quad (3.9)$$

The Equation (3.7) to (3.9) model was inspired by Pavlov et al. (2014), which uses mass balances for two control volumes and a global momentum balance. The difference

between Pavlov's model and the one used in this article is in the mass balance equation for the control volume 1. Pavlov et al. (2014) also assumes the hypothesis that there is no accumulation in the annular section. This hypothesis cannot be assumed for the system under study. Then, the mass balance in the control volume 1 will be represented by Equation (3.7). In contrast, the no accumulation hypothesis is admissible for Equation (3.8) and (3.9). Additionally, the system equations consider isothermal properties, single-phase flow, and incompressible fluid. Therefore, the compressibility factor, viscosity, and density are fixed.

Fig. 3.4 shows that the ESP system's installation is cooled by the production fluid, just like a conventional one. This configuration indicates that the heat exchange between the motor and the fluid can conflict with the isothermal system's hypotheses. Thus, in obtaining the equations, it is assumed that the variation in physical and thermodynamic properties is negligible.

Equation (3.7) gives the level of liquid in the well annular. This approach is appropriate for the ESP prototype in this analysis; however, if necessary, the equivalent intake pressure is written as follows:

$$p_{in} = L_a \rho g \quad (3.10)$$

In the current literature models, the production flow,  $q_c$  in Equation (3.8), is a function of the square root of differential pressure over the choke valve. In contrast, the reservoir flow is commonly considered ideal and given by:

$$q_r = PI(p_r - p_{bh}) \quad (3.11)$$

where  $PI$  is the productivity index,  $p_r$  is the reservoir pressure, and  $p_{bh}$  is the bottom hole pressure. However, Fig. 3.4 shows that the reservoir's behavior cannot be considered ideal and linear as in Equation (3.11) because of a valve in the reservoir pipe. Therefore, the accepted hypothesis is that the reservoir behaves like a valve, and Equation (3.7) is the adequate equation.

The global moment balance (Equation (3.9)) provides the system's flow dynamics.  $\bar{A} = \frac{A_1 + A_2}{2}$  is the average cross-section area, the average density is  $\bar{\rho} = \frac{\rho_1 V_1 + \rho_2 V_2}{V_1 + V_2}$  and the average length is  $\bar{l} = \frac{l_1 + l_2}{2}$ . Therefore, in Equation (3.9),  $\Delta p_f$  is the total pressure loss composed of the shear stress and the well's geometric restrictions. Several approaches exist to treat the friction factor, and Rønning (2011) uses the Blasius equation to obtain it. In this work, since the facility used for the experiments is a laboratory (Fig. 3.4) with low aggressiveness of fluids and a controlled environment, it was considered that the Blasius equation provides an acceptable representation of the friction loss. The friction factor values can be obtained from the following equation:

$$\begin{aligned} \Delta p_f &= F_1 + F_2, \\ F_i &= f_i \frac{\rho L_i \pi D_i}{4} \left( \frac{q_i}{A_i} \right)^2, \\ f_i &= \begin{cases} \frac{64}{Re}, & \text{if } Re < 4000, \\ 0.3164 Re^{-0.25}, & \text{otherwise,} \end{cases} \end{aligned} \quad (3.12)$$

where  $f_i$  is the friction factor,  $L$  is the length of the  $i$ -th section,  $D_i$  is the diameter,  $q_i$  is the volumetric flow in the section, and  $A_i$  is the cross-section area. The hydrostatic pressure is calculated for the two sections and is represented by

$$\Delta p_h = \Delta P_{h_1} + \Delta P_{h_2}, \quad (3.13)$$

$$\Delta P_{h_1} = \rho_1 g h_1 = \rho_1 g (h_{bh} - h_p), \quad (3.14)$$

$$\Delta P_{h_2} = \rho_2 g h_2 = \rho_2 g (h_p - h_c). \quad (3.15)$$

The differential pressure over the ESP is  $\Delta p_p$ , written as a function of the head, frequency, and average flow. The general expressions are:

$$\Delta p_p = H(f, q_b) \rho g, \quad (3.16)$$

$$H(f, q_b) = H_0(q_p, f_0) (f/f_0)^2, \quad (3.17)$$

$$H_0 = x_n q_m^n + x_{n-1} q_m^{n-1} + \dots + x_1 q_m + x_0, \quad (3.18)$$

where  $H(f, q_b)$  is the head developed by the pump, which is obtained from the use of the affinity law (Equation (3.17)) on the head in the reference frequency ( $H_0$ ). The manufacturer supplies the  $H_0$  pump polynomial for freshwater. The  $H_0$  viscosity correction proposed by **API RP 11S2** and other methods such as Turzo, Takacs e Zsuga (2000) are used to update  $H_0$  polynomial for different fluids. In this work, Turzo, Takacs e Zsuga (2000) method is used to do it.

The factory information about the installed ESP in the system indicates that Equation (3.18) can be expressed by a second-order polynomial equation, so the expression included in the model is

$$H_0 = x_2 q_m^2 + x_1 q_m + x_0. \quad (3.19)$$

Note that there is a mismatch between the desired and the real frequency of the pump operation. The real frequency is obtained in the lab using a set of accelerometers. However, this kind of instrumentation is not available in oil fields. So, to reproduce the oil field conditions, the frequency uncertainty is included in the uncertainty of the estimated parameters  $x_2$ ,  $x_1$ , and  $x_0$ .

### 3.3.3 Estimation problem and model adaptation

The equations of the previous section define the proposed dynamical model for the ESP prototype. Those equations have a set of parameters to describe the behavior of the leading equipment of the system, specifically:  $K_c$ ,  $K_r$ ,  $x_2$ ,  $x_1$ , and  $x_0$ . They form the set  $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$  respectively, obtained from applying the method in Subsection 3.2.1. The model's variables and units are summarized in Table 3.1. The other parameters and physical properties are shown in Table 3.2.

Using parameters and variables in the International System of Units (SI) units may imply distortions in some variables' magnitude. This issue's main variables are the pressure in Pascals and the flow in  $m^3/s$ . In the ESP prototype, the pressure variables have an order of magnitude of  $10^5$ , and the flow rate, in turn, an order of magnitude of  $10^{-4}$ .

Table 3.1: Model variables

	Variable	Unit
$p_r$	Reservoir pressure	$Pa$
$p_{bh}$	Bottom hole pressure	$Pa$
$p_{wh}$	Wellhead pressure	$Pa$
$p_m$	Manifold pressure	$Pa$
$q_r$	Reservoir flow rate	$m^3/s$
$q_c$	Production flow rate	$m^3/s$
$q_m$	Average flow rate in well	$m^3/s$
$F_1$	Frictional pressure drop below ESP	$Pa$
$F_2$	Frictional pressure drop above ESP	$Pa$
$H$	Head developed by ESP	$m$
$f$	ESP frequency	$Hz$
$Z_c$	Production valve opening	-
$Z_r$	Reservoir valve opening	-

For the parameters, if the pressure and flow rate are in SI units, the parameter values will also have different magnitude orders. This difference can cause numerical fluctuations during the solution of the model or parameter estimation process. To solve this issue, the model's parameters and variables are in multiples of SI units, and the output variables are modified to a normalized form.

In this new form, the pressure variables are in *bar* units, the flow rate is in  $L/s$ , and the valves' constants are in  $L/s/bar^{0.5}$ . Normalization parameters are included, and the Equation (3.7), (3.8), and (3.9) becomes:

$$\frac{dL_a^*}{dt} = \frac{1}{V_w} \left( K'_r Z_r \sqrt{(p'_r - p'_{bh})} - q'_m \right), \quad (3.20)$$

$$\frac{dp_{wh}^*}{dt} = \frac{\beta'_2}{V'_2 \Delta P_{max}} \left( q'_m - K'_c Z_c \sqrt{(p'_{wh} - p'_m)} \right), \quad (3.21)$$

$$\frac{dq_m^*}{dt} = \frac{\bar{A}}{\bar{\rho}' \bar{l} \Delta q_{max}} (p_{bh} - p_{wh} + \Delta p_p - \Delta p_h - \Delta p_f), \quad (3.22)$$

where  $V_w = AL_w \cdot 10^3$ ,  $L_a^* = \frac{L_a}{L_w}$ ,  $p_{wh}^* = \frac{p_{wh}}{\Delta P_{max}}$ ,  $q_m^* = \frac{q_m}{\Delta q_{max}}$ ,  $V'_2 = V_2 \cdot 10^3$ , and  $\bar{\rho}' = \bar{\rho}/10^3$ . The normalization parameter,  $L_w = l_1 + l_2$  is the oil well length,  $\Delta P_{max} = P_{max}^{discharge} - P_{min}^{intake}$  is the maximum differential pressure at the pump, and  $\Delta q_{max} = q_{max} - q_{min}$  is the pump's differential flow rate. The new pressure variables are  $p'_r = p_r/10^5$ ,  $p'_{bh} = p_{bh}/10^5$ ,  $p'_{wh} = p_{wh}/10^5$ ,  $p'_m = p_m/10^5$ . The flow rate is  $q'_m = q_m \cdot 10^3$  and the valve's constants are  $K'_c = K_c \cdot 10^3 \cdot 10^{0.5}$ ,  $K'_r = K_r \cdot 10^3 \cdot 10^{0.5}$ . The unit of  $\beta'_2$  in Equation 3.8 is in  $Pa$ , and in Equation (3.21) must change to  $bar$  ( $\beta'_2 = \beta_2/10^5$ ). In Equation (3.22), using pressure variables in  $Pa$  is more convenient, as the head loss terms are directly calculated in  $Pa$ .

In the prototype of Fig. 3.4, the head developed by the pump (Equation (3.17)) is not measured. This implies that to obtain the parameters  $x_2$ ,  $x_1$ , and  $x_0$ , the model

Table 3.2: Model parameters

	Parameter	Value	Unit
$h_{bh}$	Height from the reservoir to choke	32	$m$
$h_{bp}$	Height from the pump intake to choke	22.7	$m$
$l_1$	Length from the reservoir to choke	9.3	$m$
$l_2$	Length from the pump intake to choke	22.7	$m$
$r_1$	Pipe radius below ESP	0.110	$m$
$r_2$	Pipe radius above ESP	0.0375	$m$
$A$	Cross-section area of the annulus	0.033595	$m^2$
$\bar{A}$	Average cross-section area	0.0108	$m^2$
$\bar{l}$	Average length	16	$m$
$V_1$	Pipe volume below ESP	0.3286	$m^3$
$V_2$	Pipe volume above ESP	0.1003	$m^3$
$g$	Gravity constant	9.81	$m/s^2$
$\mu$	Fluid viscosity	0.00888667	$Pa.s$
$\rho$	Fluid density	855	$kg/m^3$
$\beta_2$	Bulk modulus	1.8E+9	$Pa$
$f_0$	ESP characteristics ref. freq.	60	$Hz$
$q_{min}$	Minimum flow rate	0	$m^3/s$
$q_{max}$	Maximum flow rate	1.1E-3	$m^3/s$
$K_c$	Choke valve constant	Estimated	$m^3/s/Pa^{0.5}$
$K_r$	Reservoir valve constant	Estimated	$m^3/s/Pa^{0.5}$
$x_2$	Head's polynomial coefficient	Estimated	$s^2/m^5$
$x_1$	Head's polynomial coefficient	Estimated	$s/m^2$
$x_0$	Head's polynomial coefficient	Estimated	$m$
$\Delta P_{max}$	Pressure normalization parameter	12.53	$bar$
$\Delta q_{max}$	Flow rate normalization parameter	1.1	$L/s$
$L_w$	Level normalization parameter	32	$m$

needs to infer the values of  $H(f, q_b)$ . Thus, in estimating the parameters, the optimizer manipulates  $x_2$ ,  $x_1$ , and  $x_0$  to adjust the model outputs. This procedure differs from the traditional parameter estimation approach in which measurements are required.

### 3.3.4 Evaluation of uncertainty of experimental data

The set of experimental data obtained from the prototype included 101 hours of experiments. The system's settling time is 2 hours; in other terms, the time that the system takes for the response to a step-type input to be between  $\pm 5\%$  of the steady-state value. Since the value of three times the settling time was adopted, the input variables of the system, frequency ( $f$ ), and the choke valve opening factor ( $Z_c$ ) were kept constant by six hours for each steady state. In this way, the data stand for 18 steady-state and 15 dynamic regions.

The uncertainty of the experimental data needs to be evaluated in terms of all steady

states of the ESP prototype. The uncertainty analysis is made by the method and the hypotheses discussed in Subsection 3.2.2. As a way of exemplifying the investigation, Table 3.3 shows the results for the following operating point:

- Frequency of motor pump operation at 45 Hz.
- Choke valve opening at 100%.

In Table 3.3, the standard type A uncertainty is represented by the mean, standard deviation, and variance. The standard type B uncertainty represents the instrument resolution, precision, and instrumental drift. These type B uncertainty components have been obtained from the calibration certificates and instruments data sheet. The combined uncertainty ( $u_c$ ) is obtained by Equation (3.6). The Welch–Satterthwaite equation obtains the Effective degree of freedom ( $\nu$ ) (BIPM et al., 2008). The coverage factor  $k(\nu)$  is a value based on the  $t$ -distribution for the degrees of freedom to a 95 % confidence level. The expanded uncertainty is  $U = ku_c$ . Moreover, the relative uncertainty is the expanded uncertainty divided by the mean.

Table 3.3: Uncertainty evaluation results

Variable	Sensor limit		Type A uncertainty			Type B uncertainty				Combined uncertainty				
	Min	Max	Mean	Standard deviation	Variance	Resolution	Precision	Instrumental drift	Combined Type B	Combined uncertainty	Effective degree of freedom	Coverage factor	Expanded uncertainty	Relative uncertainty
Reservoir pressure/(bar)	0.000	10,340	2.170	0.010	9.199E-05	0.030	0.030	0.010	0.037	0.039	24,761	2.061	0.080	1.783%
Bottom hole pressure/(bar)	0.000	10,340	1.564	0.002	2.355E-06	0.030	0.030	0.010	0.037	0.037	22,019	2.074	0.078	2.398%
Wellhead pressure/(bar)	0.000	103,420	5.528	0.007	4.283E-05	0.030	0.210	0.180	0.275	0.275	20,057	2.086	0.574	4.978%
Manifold pressure/(bar)	0.000	4,000	0.247	0.008	7.087E-05	0.030	0.010	0.100	0.031	0.032	14,665	2.136	0.068	12.893%
Production flow/(m <sup>3</sup> /h)	0.000	4,500	1.941	0.001	1.151E-06	0.030	0.010	0.001	0.030	0.030	12,000	2.179	0.066	1.560%
Reservoir flow/(m <sup>3</sup> /h)	0.800	13,000	1.697	0.014	1.969E-04	0.030	0.030	0.030	0.047	0.049	34,728	2.031	0.099	2.885%
Intake Level/(m)	-	-	18.542	0.019	3.448E-04	-	-	-	-	0.447	22,019	2.074	0.927	4.837%

The analysis of Table 3.3 shows that the dominant contribution is Type B uncertainty. This influence indicates information about how the measuring system increases errors and is more important than measurement repeatability. Generally, measurements of values close to the detection limits will have more significant uncertainty. The precision value shows this behavior as it is a function of the detection limits of the instruments. The analysis of the other stationary states will produce similar results.

### 3.3.5 Parameter estimation and validation

As discussed in Section 3.2, the validation method includes dividing the sample data set into two subsets. Thus, the largest is used for parameter estimation and validation processes and contains about 70 hours of data. The other set for test purposes contains approximately 31 hours' worth of data.

Fig. 3.5 shows the excitation signals at the pump operating frequency and the choke valve's opening in the first data set. These disturbances in the prototype provide information about the different dynamic regimes and steady states.

Supplementary to experimental data, the MCMC simulation requires an initial guess for the parameter estimation. The initial guess for the pump curve can be the factory curve with the Turzo, Takacs e Zsuga (2000) correction for oil. The valve parameters are obtained from the respective datasheet. However, it is possible to find a better estimate

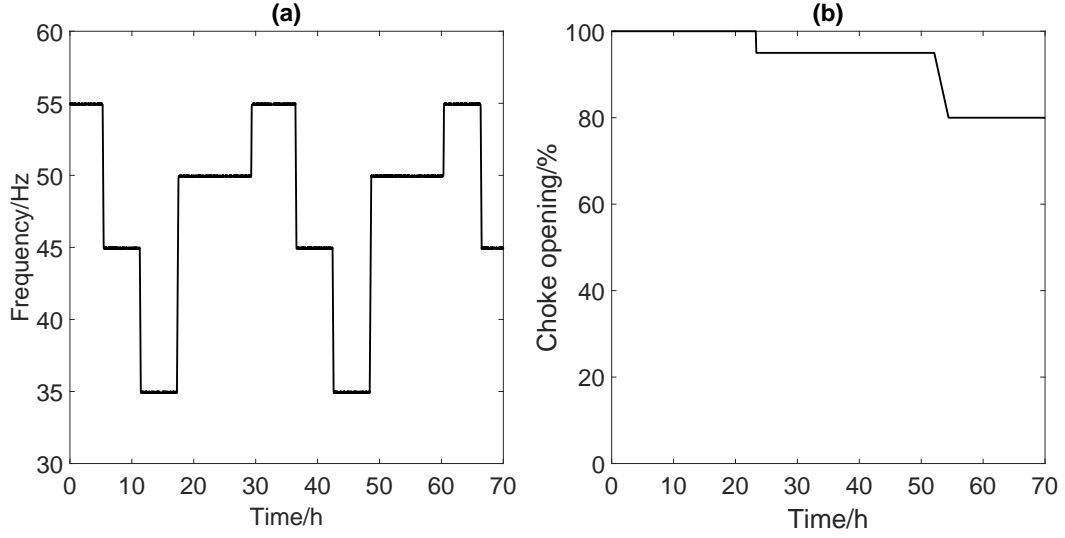


Figure 3.5: Inputs signal inserted in the prototype.

by solving an optimization problem. One way of doing so is to find the minimum point of the likelihood function through a gradient method. This optimization process provides the following set of parameters, shown in Table 3.4.

Table 3.4: Initial guess estimation for the parameters.

Parameter	$\theta_1/(Ls^{-1} \cdot bar^{-1})$	$\theta_2/(L \cdot s^{-1} \cdot bar^{-1})$	$\theta_3/(s^{-2} \cdot m^{-5})$	$\theta_4/(s/m^{-2})$	$\theta_5/m$
Value	0.2261	0.6869	-95.4098	6.9056	177.7124
Minimum	0.1131	0.435	-47.7049	3.4528	142.1699
Maximum	0.3392	1.0304	-143.1147	10.3584	213.2548

The maximum and minimum values in Table 3.4 are the limits imposed in the MCMC algorithm as a search region. These constraints are necessary to guarantee that the algorithm does not converge to unwanted regions.

The curves of Fig. 3.6 show the sensitivity analysis for the likelihood function in the desired region of convergence. In Fig. 3.6, graphs (a), (b), and (c) represent the influence of each of the system's outputs: level, pressure in the choke, and average flow, in the value of likelihood, respectively. In this figure, the parameters  $\theta$  were transformed to the dimensionless form concerning the values of Table 3.4, to equalize the visualization.

A closer analysis of Fig. 3.6 shows that the parameters influence the likelihood function differently. In particular,  $\theta_4$  has a minor influence on the value of the function since there is no local minimum for it. However, Fig 3.6 shows that the region limited by the values in Table 3.4 has a local minimum.

After defining the initial estimation of the parameters, the solution to the problem proposed in Subsection 3.3.3 can be found. The solution was found with the DRAM algorithm proposed by Haario, Saksman e Tamminen (2001b) and Haario et al. (2006a).

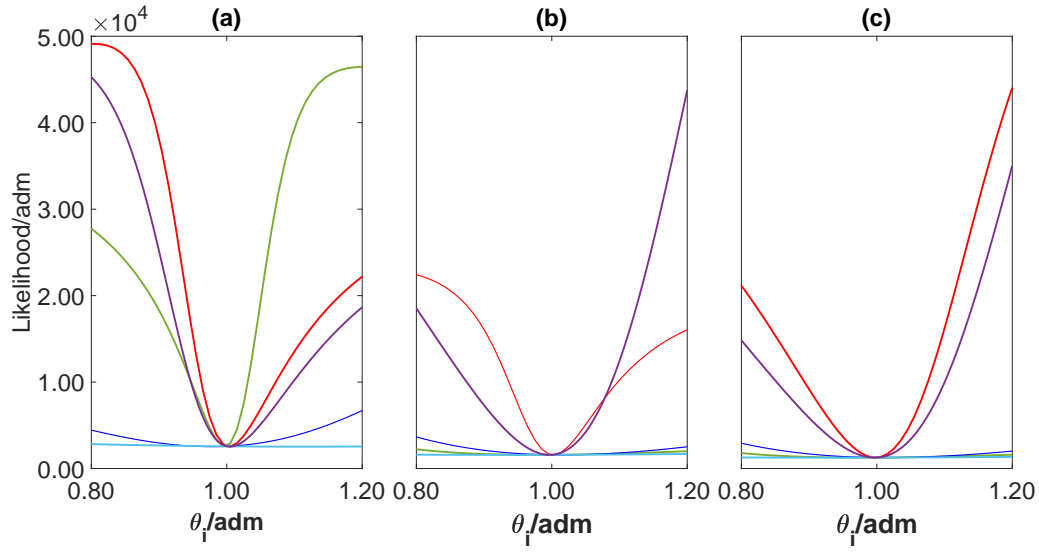


Figure 3.6: Sensitivity analysis of the likelihood function.

(a)  $L_a$  contribution; (b)  $p_{wh}$  contribution; (c)  $q_m$  contribution; (—)  $\theta_1$ ; (—)  $\theta_2$ ; (—)  $\theta_3$ ; (—)  $\theta_4$ ; (—)  $\theta_5$ .

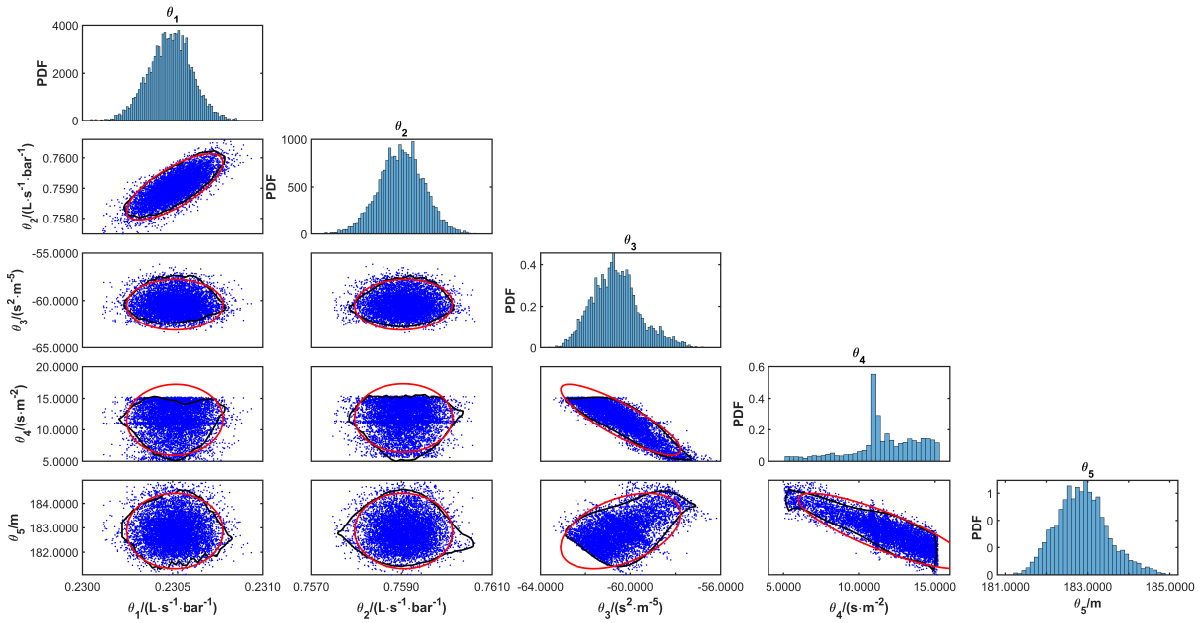


Figure 3.7: Confidence bound of estimated parameters.

(•) Parameter chosen by the MCMC method; (—) Gaussian region; (—) Possolo (2010) region.



The MCMC simulation was performed to obtain a set of 20,000 samples of  $\theta$ , excluding 1,000 points of burn-in, and the results are presented in Table 3.5 and Fig. 3.7. The total number of  $\theta$  samples tested was 57,366, so the algorithm discarded 37,366 samples during the solution process to obtain the final PDF.

Table 3.5: MCMC Statistics

	Mean	Expectance ( $\hat{\theta}$ )	Standard deviation	MCMC error
$\theta_1/(Ls^{-1} \cdot bar^{-1})$	0.2283	0.2284	0.0005	1.5620E-5
$\theta_2/(Ls^{-1} \cdot bar^{-1})$	0.7221	0.7224	0.0006	1.6433E-5
$\theta_3/(s^{-2} \cdot m^{-5})$	-75.3973	-74.5521	0.0143	0.0013
$\theta_4/(s/m^{-2})$	7.8962	7.5521	0.2264	0.0245
$\theta_5/m$	180.7896	180.7844	0.0035	0.0003

The analysis of Fig. 3.6 shows that the likelihood function dependency of the parameters  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , and  $\theta_5$  is accentuated. This dependency means that the acceptance of new points by the MCMC is small and converges to a close region, as seen in Fig. 3.7. In turn, the MCMC standard deviation and the error values, shown in Table 3.5, agree with the behavior observed in Fig. 3.6 and 3.7 because the MCMC standard deviation alters the spread observed in Fig. 3.6 and 3.7. Table 3.5 also includes the average and nominal values of the parameters represented by the expected value (most likely value).

As seen above, parameter 4 is more difficult to converge. A possible solution would be to represent the pump curve (Equation (3.18)) solely by  $\theta_3$ , and  $\theta_5$ . This approach would not remove the pump performance curve's parabolic shape and simplify the parameter estimation process. However, the current pump literature commonly presents complete polynomials, and the minor interference in the model response allowed this parameter to be left in Equation (3.19).

The confidence bound of the parameters for 95% of the confidence level is shown in Fig. 3.7 by the black and red lines. This region is generally built to hypothesize that the parameters have a Gaussian distribution (red lines). However, this hypothesis is invalid for constructing most parts involving the  $\theta_4$  and  $\theta_5$ . In these cases, Possolo (2010) provides a method to build these regions that will result in the most suitable confidence bounds (black lines).

### 3.3.6 Validation

The model prediction's confidence bound can be obtained with the parameters in the PDF. Fig. 3.8, 3.9, and 3.10 show the comparison between the model's prediction and the experimental data. The blue circles in Fig. 3.8, 3.9, and 3.10 are the experimental data obtained from the ESP system of Fig. 3.4. The prediction of the model for the most likely value of the parameters is the black line. Fig. 3.8, 3.9, and 3.10 also show two shadow areas. The blue area is the experimental's confidence bound and was built with the methods presented in Section 3.2.2. The grey area represents the confidence bound of the prediction of the model. In both cases, the confidence level used is 95%.

The first part of the validation is to compare the system's dynamic behavior. Fig. 3.8, 3.9, and 3.10 show that the model's dynamic behavior has the same trends as the real system. The phenomenological model, combined with the estimated parameters, seems to be a way to validate the models from a dynamic perspective. The model's prediction behavior is also consistent with the whole system behavior.

The propagation of the PDF allows obtaining the dynamic uncertainty of the model. The gray shaded areas in Fig. 3.8, 3.9, and 3.10 show the model's prediction's dynamic uncertainty. In contrast, the evaluation of the experimental data's uncertainty by the GUM method uses a steady-state hypothesis. Therefore, the uncertainties are compared when the experimental data are steady.

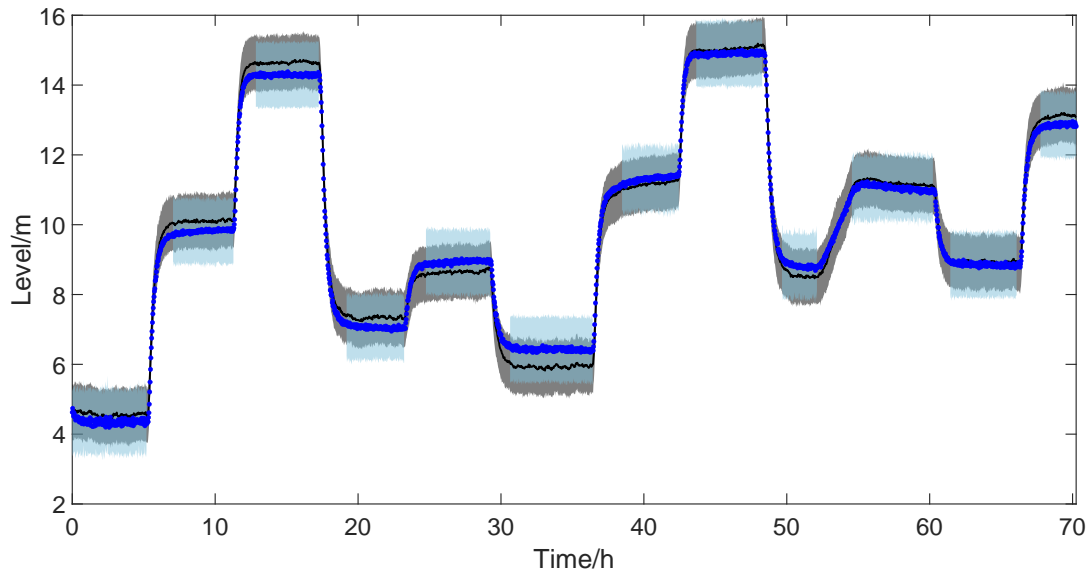


Figure 3.8: Intake level - direct validation.

(—) Model response for the nominal set of  $\theta$ ; (•) Experimental data; ■ Confidence bound of the prediction; ■ Confidence bound of the experimental data.

Therefore, the superposition of the confidence bounds of the prediction and the experimental data, as shown in Fig. 3.8, 3.9, and 3.10, indicates that the model can describe the experimental data, given the levels of uncertainty presented. In other words, the model is considered validated and adequately represents the system's dynamic behavior.

### 3.3.7 Test

In addition to the validation presented in the previous subsection, it is necessary to evaluate the system's behavior against experimental data that were not used to improve the model's fit. Fig. 3.11 shows the signal inputs of the cross-validation data. The new scenario has five distinct dynamic sections and steady states. The inputs change asynchronously, but in the third input change, both inputs change simultaneously. Fig. 3.12, 3.13, and 3.14 compare the model's prediction with the separate data set for cross-

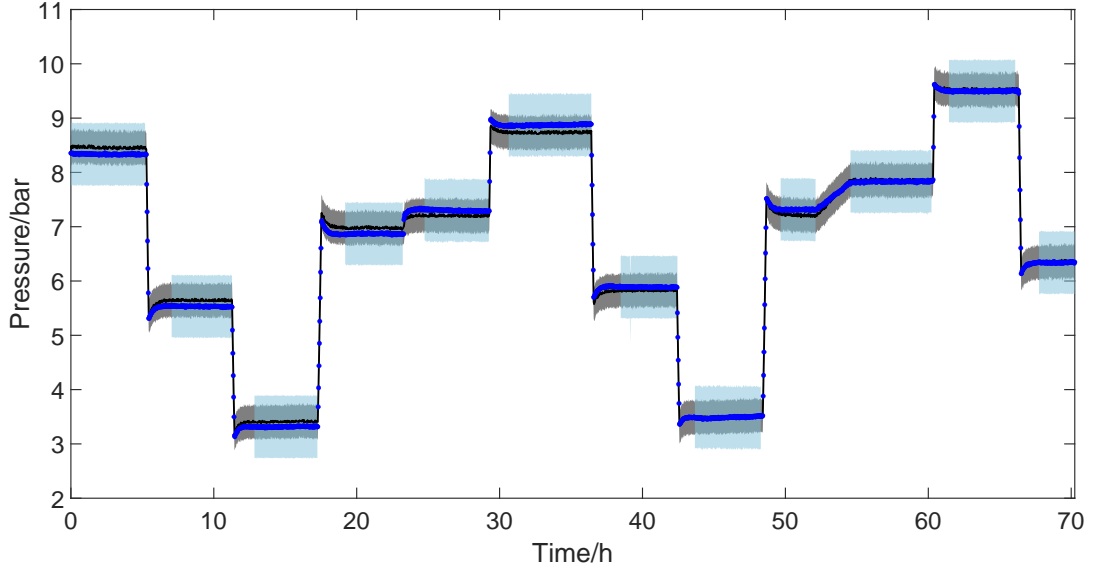


Figure 3.9: Wellhead pressure - direct validation.

(—) Model response for the nominal set of  $\theta$ ; (•) Experimental data; ■ Confidence bound of the prediction; ■ Confidence bound of the experimental data.

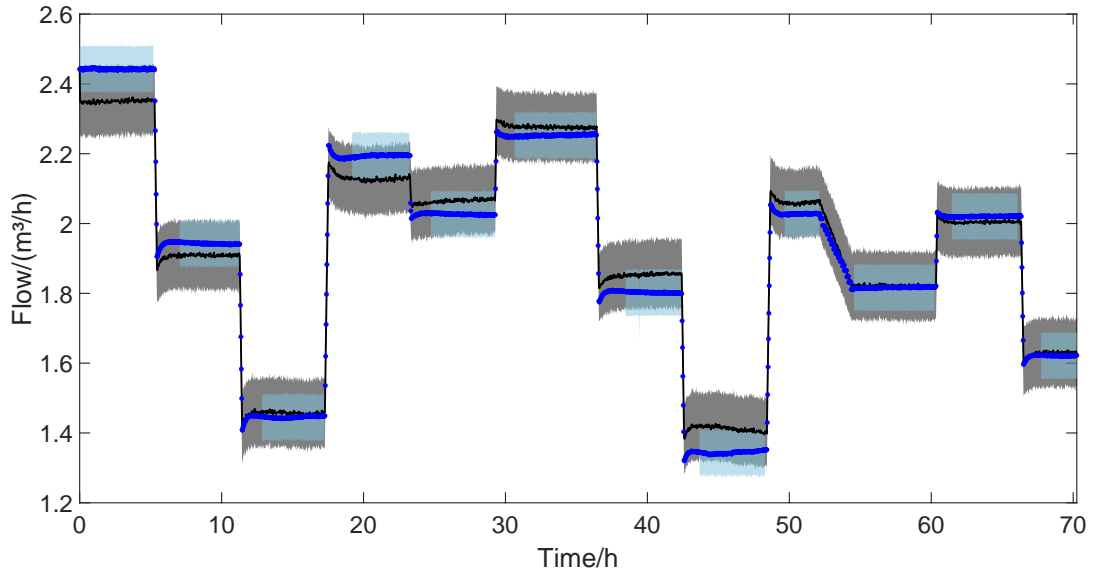


Figure 3.10: Production flow - direct validation.

(—) Model response for the nominal set of  $\theta$ ; (•) Experimental data; ■ Confidence bound of the prediction; ■ Confidence bound of the experimental data.

validation. The overlap of the confidence bounds indicates that the model can track these experimental data sets. This result is similar to that of the direct validation. Thus, the results of cross-validation corroborate those observed in Fig. 3.8, 3.9, and 3.10.

Therefore, the method proposed in this article allows for obtaining validated ESP system models. Specifically, the Bayesian approach constructs the parameters' PDF and the regions covered by the experimental data and the model. It is considered, however, that traditional methods of inferring model uncertainty failed for the ESP model, as it has a stiff behavior, and the derivative of the maximum likelihood function concerning the parameters is very sensitive to perturbation methods.

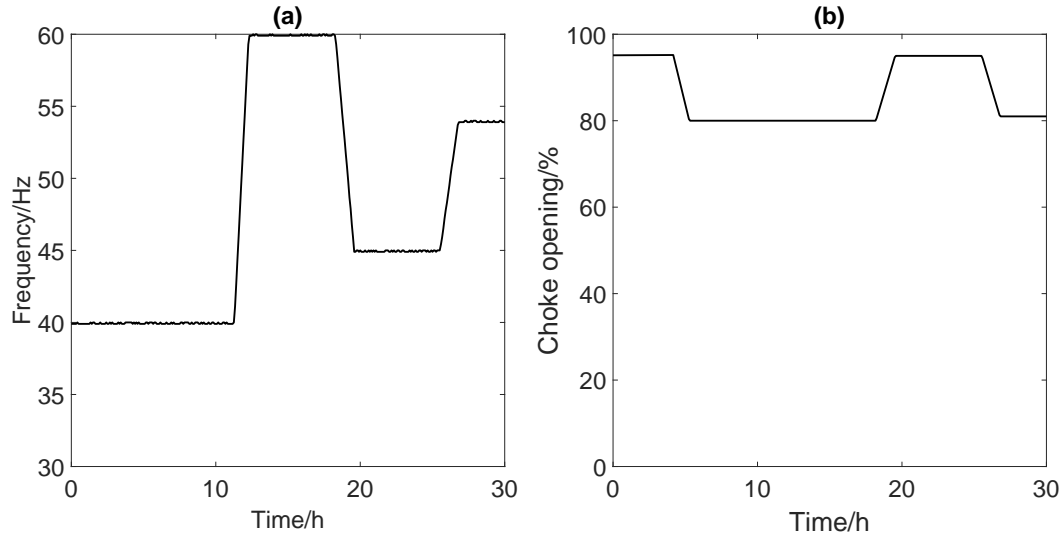


Figure 3.11: Inputs signal inserted in the prototype to test dataset.

### 3.4 CONCLUSION

This chapter presented a Bayesian approach to validating ESP system models. The proposed method uses data from sensors installed to update the equipment's operating parameters. The benefits are obtaining parameters in an integrated manner and systematically treating uncertainty.

The method obtains the combined PDF of the model parameters and the prediction's confidence bound through Bayesian inference. In parallel, the GUM method is standardized to assess the uncertainty of the experimental data. The validation is achieved in two steps. The first involves comparing the dynamic behavior of the model's prediction with the experimental data. The second is in steady-state to identify whether there is an overlap in the model predictions' confidence bounds and the experimental data.

The results indicate that the method effectively builds the joint PDF of the model parameters and obtains the confidence bounds. Furthermore, the validation shows that the model can represent the experimental data within the model's existing uncertainties. Therefore, the resultant model is a candidate for use as a surrogate model for the experimental apparatus.

Future work may be carried out toward constructing and validating multiphase and

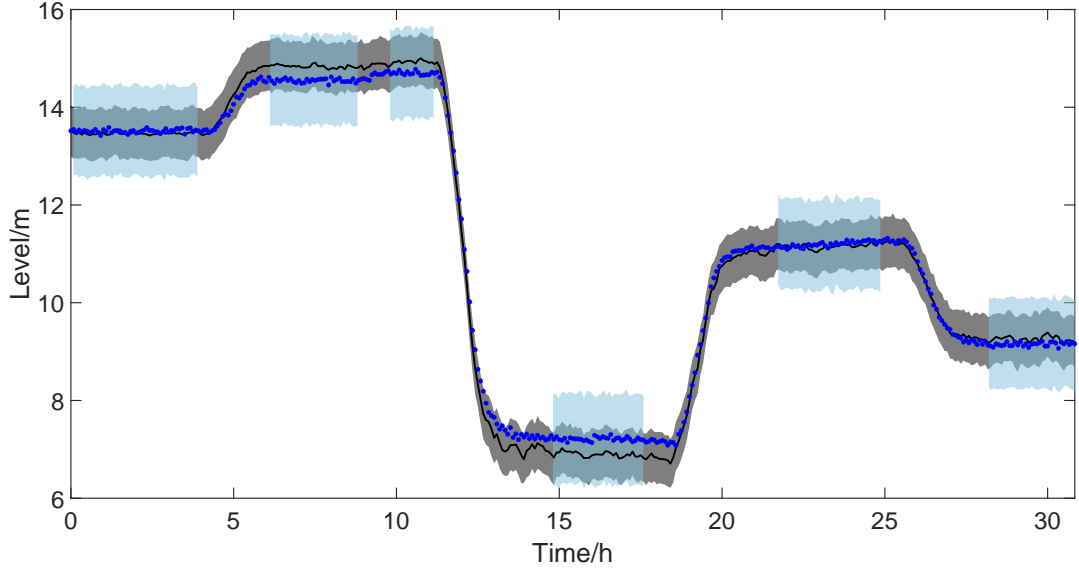


Figure 3.12: Production flow in the test dataset.

(—) Model response for the nominal set of  $\theta$ ; (•) Experimental data; ■ Confidence bound of the prediction; ■ Confidence bound of the experimental data.

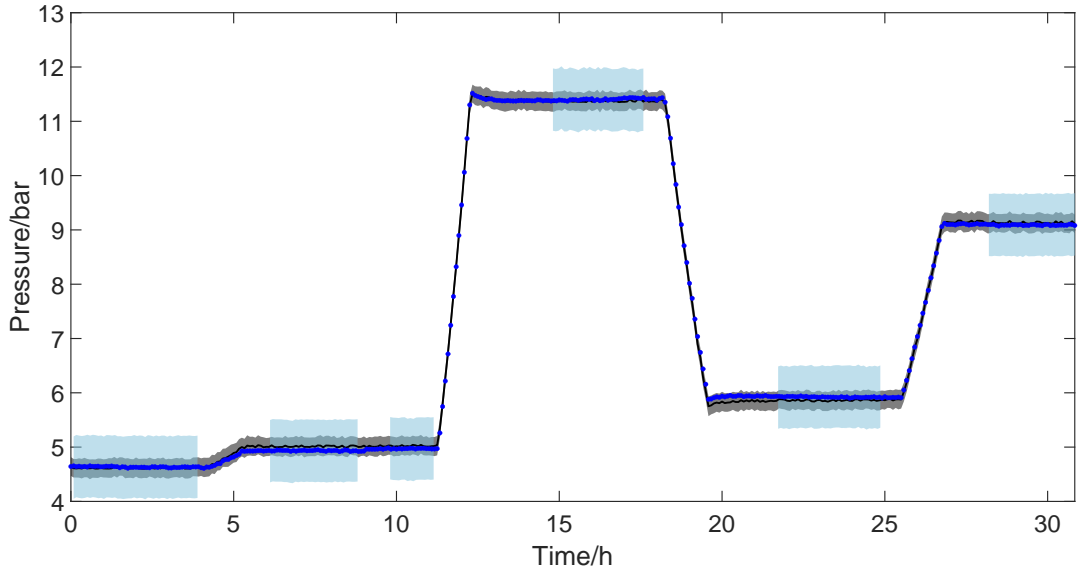


Figure 3.13: Wellhead pressure in the test dataset.

(—) Model response for the nominal set of  $\theta$ ; (•) Experimental data; ■ Confidence bound of the prediction; ■ Confidence bound of the experimental data.

non-isothermal models.

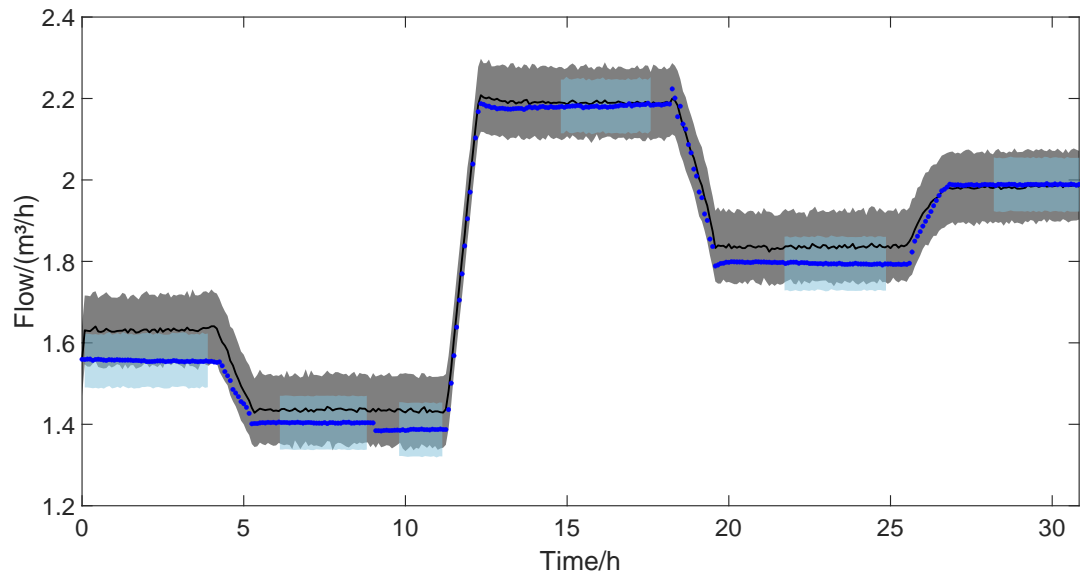


Figure 3.14: Production flow in test dataset.

(—) Model response for the nominal set of  $\theta$ ; (•) Experimental data; ■ Confidence bound of the prediction; ■ Confidence bound of the experimental data.

## CHAPTER 4

# AN UNCERTAINTY APPROACH FOR ELECTRIC SUBMERSIBLE PUMP MODELING THROUGH DEEP NEURAL NETWORK

Published on **Costa, E. A.**; Rebello, C. M.; Santana, V. V.; Reges, G.; Silva, T. O.; Abreu, O. S. L; Ribeiro, M. P.; Foresti, B. P.; Fontana, M.; Nogueira, I. B. D.; Schnitman, L. 2024. An uncertainty approach for Electric Submersible Pump modeling through Deep Neural Network. Heliyon. DOI: <https://doi.org/10.1016/j.heliyon.2024.e24047>

### 4.1 INTRODUCTION

This chapter proposes a new methodology to identify and validate deep learning models for artificial oil lift systems that use electric submersible pumps. This Chapter tests hypotheses 1, 2, and 3, presented in section 1.1, in the context of training, validation, and uncertainty assessment of artificial intelligence models. The proposed method allows for the acquisition of models and the simultaneous and systematic uncertainty assessment of the prediction. The technique employs a nonlinear model to generate training and validation data. This model was used to overcome the limitations of the need for data for training deep learning models. The developed models are validated against experimental data after training and validation with synthetic data. Validation is also performed using the models' uncertainty assessment and experimental data. The results show that the proposed methodology obtained models representing both the nonlinear model's dynamic behavior and the experimental data. Additionally, the uncertainty of the generated deep learning models has the same order of magnitude as that of the nonlinear model. This uncertainty assessment shows that the built models were adequately validated. The proposed deep learning models can be applied in several applications requiring a reliable and computationally lighter model. Hence, the obtained AI dynamic models can be employed for digital twin construction, control, and optimization.

### 4.2 METHODOLOGY

Figure 4.1 presents the method scheme used in this Chapter. All process is divided into eight stages. The first stage consists of data curation. The second stage is generating synthetic data for the models' identification. After, DNN network architecture is estimated by hyperparameters optimization using the Hyperband method (LI et al., 2016). The uncertainty assessment of experimental data is the fourth stage and follows the settled in BIPM et al. (2008) (Guide to the expression of uncertainty in measurement). In turn, the MCMC training process proposed here is the fifth stage in which the training data and the hyperparameters are grouped to generate the Probability Density Function

(PDF) of the weights and biases of the DNN models. The last steps are inferring the network's uncertainty, propagation to the model's outputs, and testing with simulated and experimental data. The case study in the Experimental Procedures section will provide more details about each step of the methodology, presenting the preliminary results for the case study.

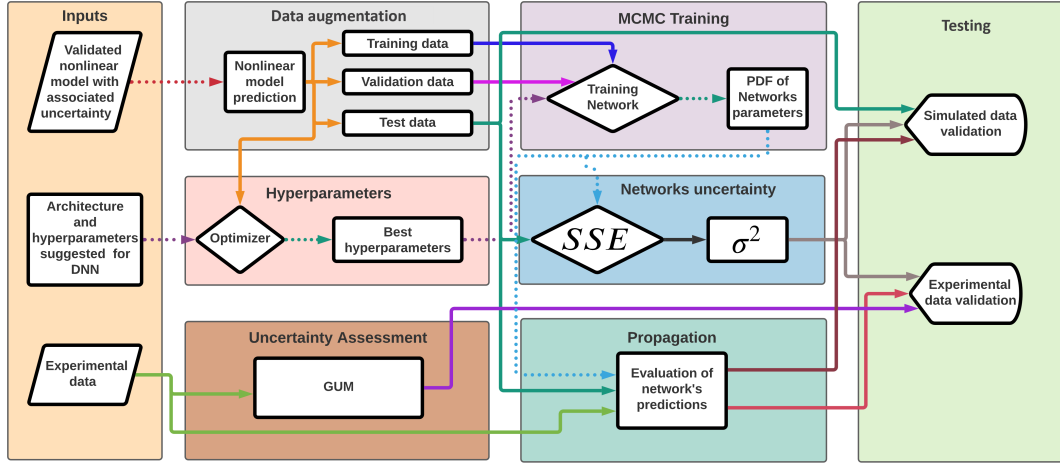


Figure 4.1: Schematic diagram of the proposed method. Dotted lines are models or parameters, and solid lines are data.

## 4.3 CASE STUDY

### 4.3.1 Data augmentation

This Chapter uses the same ESP system presented and mathematically defined in Subsection 3.3.1. The ESP system is represented mathematically by three nonlinear differential equations and several algebraic equations proposed and validated in Chapter 3. In this Chapter, the ESP model is used in the proposed methodology to generate data for training and validating neural networks to represent the three process variables. This phenomenological model can provide a reliable data source to identify the proposed empirical models. This rigorous model provides the AI models with information regarding system phenomenology. This is a more cost-effective way to leverage the experimental apparatus, amplifying the data available through the mechanical model.

Figure 3.4 shows that the reservoir pressure is dependent on the level of the T-101 tank. This behavior is a stint in generating the input signals because the liquid level in the tank depends on the produced volume, hence depending on  $f$  and  $Z_c$ . However, it is possible to generate a transfer function model to represent the behavior of the reservoir pressure as a linear function of motor frequency and the choke opening. These linear functions only generate dependent signals between reservoir and manifold pressure and do not interfere with the system's non-linear behavior. The best-estimated models by the method of Ozdemir e Gumussoy (2017) are:



$$\frac{P_r(s)}{F(s)} = \frac{(-1,526 \cdot 10^{-6}s - 1,408 \cdot 10^{-10})e^{-1s}}{s^2 + 1,017 \cdot 10^{-3}s + 1,022 \cdot 10^{-8}} \quad (4.1)$$

$$\frac{P_r(s)}{Z(s)} = \frac{(-7,763 \cdot 10^{-4}s^3 - 1,298 \cdot 10^{-7}s^2 - 1,166 \cdot 10^{-11}s - 1,360 \cdot 10^{-16})e^{-1s}}{s^4 + 1,332 \cdot 10^{-3}s^3 + 2.024 \cdot 10^{-7}s^2 + 1.744 \cdot 10^{-11}s + 2.633 \cdot 10^{-16}} \quad (4.2)$$

where  $s$  is the complex Laplace variable that has a unit of frequency in  $seconds^{-1}$ ,  $F(s)$  and  $Z(s)$  are the Laplace representation of ESP motor frequency and choke valve opening level. So the model of Eq 4.1 and 4.2 relates  $P_r$  with  $f$  and  $Z_c$ .

Chapter 3 presented an experimental validation of the non-linear model for the system in Figure 3.3.1 with the uncertainty assessment of the model parameters. To assess the uncertainty of the parameters, the method of Chapter 3 considered that the uncertainty was associated with the parameters of the pump curve, the reservoir, and the choke valve. With the evaluation of the uncertainty of these parameters, Chapter 3 provided the PDF of the pump curve and flow index parameters of the reservoir and the choke valve. Also, Chapter 3 showed that the model was validated and could be used for other applications.

In this sense, in this chapter, the idea is to generate data to identify the empirical models considering the propagation of the uncertainty of nonlinear models, as seen in Figure 4.2. The goal is to create a set of  $m$  representative nonlinear models ( $y_{1,2,...,m}$ ) by sorting a sample of the PDF of the ESP model parameters provided on Chapter 3. Thus, a matrix  $\Theta_{(1,2,...,m)}$  containing each set of possible values for the model provides a set of equivalents models. This set of models allows the assessment of the dynamic uncertainty of the nonlinear model.

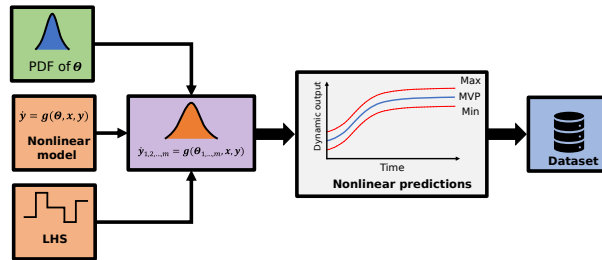


Figure 4.2: Uncertainty propagation on the nonlinear model.

A Latin hypercube sampling (LHS) signal is used as a disturbance in the phenomenological models and generates its corresponding dynamic uncertainty. In addition, these perturbations generate a dynamic dataset for training and validating the DNN models. The nonlinear model responses allow for each time sample's most likely, maximum, and minimum values. Then, the most likely trajectory is stored in the training database. It is essential to highlight that the greater the number of good possible models, the better these output responses represent the nonlinear model uncertainty. The generated dataset is divided into three subsets for cross-validation (HAYKIN, 1999). One set is

used to train the DNN models. Another is used to validate the trained models during the training process. The last set is separated to test the models after the training.

In this way, it is possible to couple Eq 4.1 and 4.2 with the model of Costa et al. (2021) to generate all the datasets. The pseudo-random binary sequence (PRBS) signal is generated for  $f$ ,  $Z_c$ , and  $p_m$  from an LHS in the first step. It used PRBS signals to a frequency between 30 and 60 Hz, choke opening between 0 and 100 %, and manometric manifold pressure between 0 and 0.2 bar. Figure 4.3 shows all input steps. The corresponding dynamics comprise a dataset with over 400 hours of experiments and 15000 samples with a time step of 100 seconds. Also, the most likely value is obtained by simulating the model response to 1000 sorted parameter sets from the PDF of Costa et al. (2021).

In this work, the generated data was divided into three subsets: one with 70 % of the data and two with 15 %. The first is used to train the DNN models, the second is used to test the chosen weights during the training process, and the last is stored in the cross-validation step.

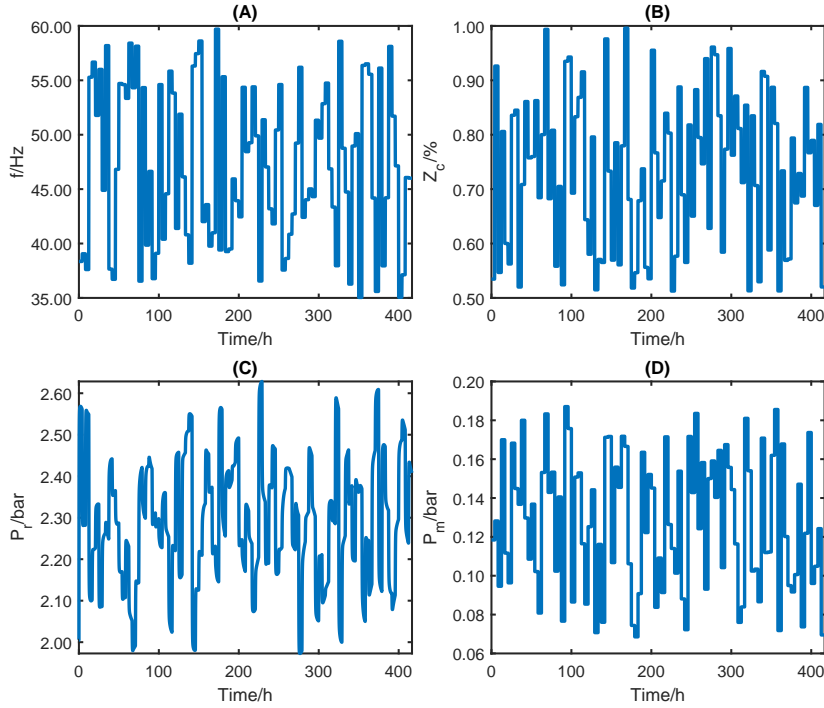


Figure 4.3: PRBS signal inputted in the nonlinear model.

#### 4.3.2 Data curation

This Chapter proposes to use a function approximator defined through a NARX using a DNN composed of a Feedforward Neural Network (FNN) with Dense layers. The

predictor that represents this structure is written as (HAYKIN, 1999):

$$y_{n+1} = F(y_n, y_{n-1}, y_{n-2}, \dots, y_{n-q+1}, u_{n-1}, u_{n-2}, u_{n-3}, \dots, u_{n-p+1}) \quad (4.3)$$

where  $F$  is a nonlinear function, the number of time-unit feedback inputs and outputs gives the model's order. These are hyperparameters related to the predictor structure. Additionally, the model sees  $n - q + 1$  steps of outputs and  $n - p + 1$  inputs. The Lipschitz order index proposed by He e Asada (1993) method is a way to determine  $q$  and  $p$ , which defines the model's order.

Figure 4.4 shows the results of the order analysis of the predictor used in this work. The number of past inputs and outputs ranges from 1 to 8. It is possible to see that the Lipschitz index is more sensitive to the lags in the outputs than in the inputs. It is also possible to observe that choosing values greater than 1 for the input order of the model and 2 for the outputs does not significantly reduce the Lipschitz index. However, this work adopted the orders as 6 for the inputs and 2 for the outputs. This choice aims to meet the requirements of the method of He e Asada (1993) and ensure that the database has enough recurrent information to train the models.

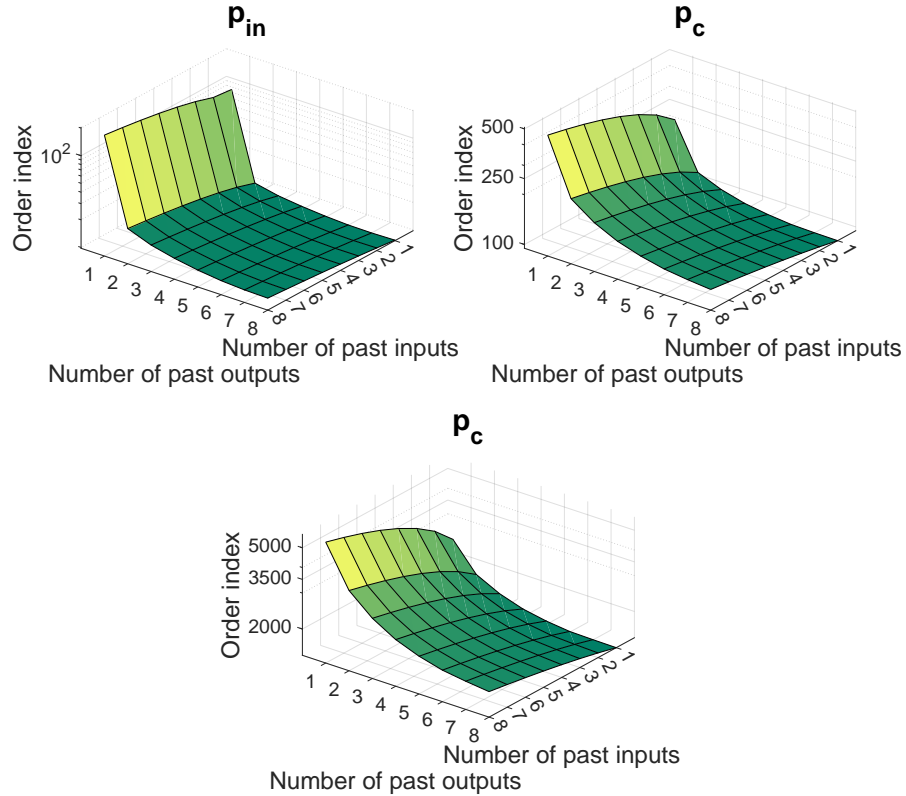


Figure 4.4: Lipschitz Quotients for identification of the Inputs and Outputs order.

It is necessary to highlight that the extrapolation capacity of a DNN model significantly depends on the provided data. The amount of experimental data available, even in the case of pilot units, needs to be increased. In this sense, using rigorous and validated

simulators to generate this training data is justified. In addition, it is noted that with experimental data, it is not always possible to cover the entire extension of the operating region since, in real plants, performing experiments with PRBS signals to construct deep learning models interferes with production interests and the process.

### 4.3.3 DNN hyperparameters and training

The optimal settings of DNN model parameters, the options of the training algorithm, and the structural parameters of the DNN are found by hyperparameters optimization. To find the DNN parameters, the Li et al. (2016) Hyperband algorithm is used through the Keras Tuner open-source toolbox available by O'Malley et al. (2019). The hyperparameters estimated were: the number of layers, activation function, number of neurons per layer, and learning rate. Table 4.1 shows the Hyperband algorithm search space configuration.

Table 4.1: Setup of the Hyperband algorithm.

Parameter	Search space
Type of layer	Dense
Number of layers	1-5
Output layer	1
Activation function	"relu", "tanh" or "linear"
Neurons per layer	10 to 60 with step 5
Dropout layer rate	0.25
Learning rate	0.0001 to 0.01 with log sampler
Metrics	Mean Absolute Error (MAE)
Loss	Mean Squared Error (MSE)
Optimizer	Adam

The resulting values provided by the Hyperband algorithm are shown in Table 4.2 and are identified for each output.

Following the methodology, the DNN model is trained with the architecture model of Table 4.2. Figures 4.5-4.7 show the Mean Absolute Error (MAE) and Mean Squared Error (MSE) for the three  $p_c$ ,  $p_{in}$ , and  $q_c$  DNN models training, respectively. The upper graphs of Figures 4.5-4.7 show the MSE and MAE values for the training data. Also, the below graphs of Figures 4.5 to 4.7 the MSE and MAE values for the test data; in other words, these parameters indicate the convergence of the adjustment of the model with a different dataset used in training, see Figure 4.5. Additionally, two stopping criteria were used. The first is the number of epochs fixed at 300. The second was an early stop option on the validation MAE value that turns off training if there is no significant decrease in the MAE value in the last 100 previous samples. However, the MSE and MAE for the validation dataset is not shown in the figure.

The convergence presented in Figures 4.5-4.7 is essential to ensure that the architecture of the DNN model is consistent with the data. If the MSE value for the training

Table 4.2: Results found by the Hyperband algorithm.

Hyperparameter	$p_{in}$	$p_c$	$q_c$
Number of layers	2	3	5
Activation function	{linear, linear} <sup>1</sup>	{linear, linear, linear}	{linear, linear, linear, linear, linear}
Neurons per layer	{55, 30}	{60, 20, 25}	{45, 60, 35, 10, 35}
Initial learning rate	$3.645 \cdot 10^{-4}$	$1.321 \cdot 10^{-3}$	$4.900 \cdot 10^{-4}$
Trainable parameters	3196	3391	6891

<sup>1</sup> Although this network is considered shallow, as it does not have many hidden internal layers, the nomenclature was maintained to avoid confusion.

data is not decreasing, the network training is not converging. On the other hand, the value of the test data can vary and it also indicates the convergence of the training.

From the point of view of training convergence, it is also essential to analyze the MAE and MSE values when the amount of data used in training the network varies. Then, each network shown in Table 4.2 was evaluated by increasing the set in 100 experiments starting at 100 until reaching 10490, which contains all train data.

Additionally, the networks were retrained in 25 replicates to analyze the average behavior. Figures 4.8 to 4.10 shows the average of the 25 MAE and MSE values for the training dataset, in which it is possible to observe that when few data are included, the training does not converge. However, the value drops dramatically when the dataset has more than 500 experiments, approximately. Figures 4.8 to 4.10 also show the MAE and MSE values when the trained DNN is evaluated with the test data. It is possible to see that convergence occurs for both data sets, and this behavior is coherent since the training network convergence is always achieved when data is added.

It is crucial to consider that the more data is included in the training, the greater the computational effort; however, the trained model will be better if more data is available. In this Chapter, however, the option was to include a greater amount of data to guarantee that the models had enough information about the entire region of operation of the system. As shown in Figure 4, the system has four exogenous variables that interfere with its behavior, each with an operating range. Thus, although training datasets with 10490 lines have a higher computational effort than sets of 2000-5000 lines, it is essential to ensure that the DNN models know the entire operating region.

#### 4.3.4 MCMC training

This Chapter proposes to use MCMC simulation to obtain the posterior PDF of the ESP DNN model parameters. Figure 4.11 shows how the PDF of DNN network parameters is obtained by sampling aleatory values from the proposal distribution. Also, the MCMC method provides the optimal parameter value and the associated variance.

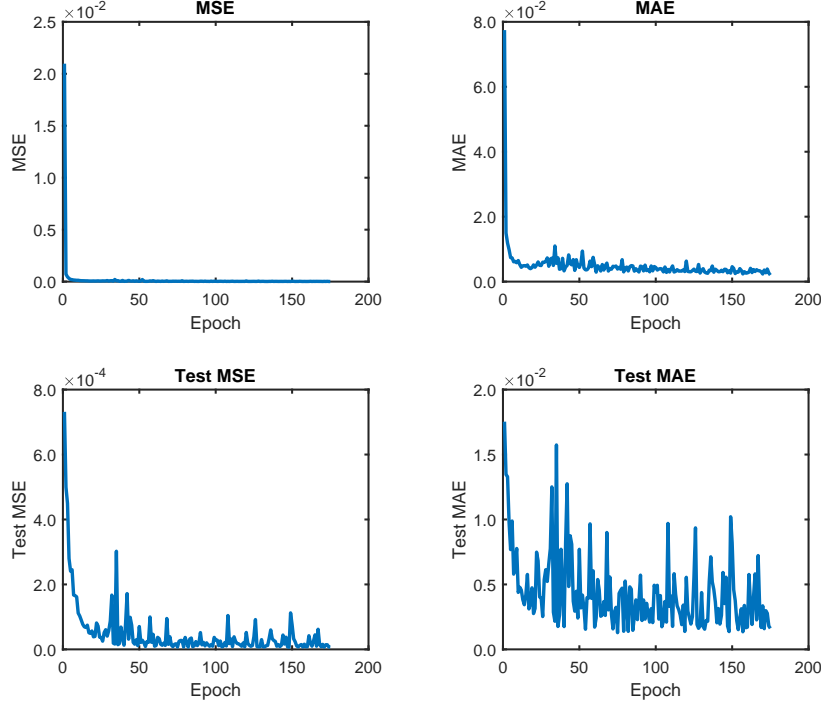


Figure 4.5: Choke pressure DNN MAE and MSE training and test values.

The Bayesian method involves obtaining the posterior probability density function (PDF) of the parameters ( $\mathbf{g}_\theta$ ), which is a function of the experimental data ( $D$ ) and prior information about the system ( $I$ ). The relationships between these variables are expressed as shown in Figure 4.11.

In this context,  $\eta$  represents aleatory values of  $\theta$ , which are the weights and biases of the DNN models. The likelihood function ( $L$ ) describes the residuals between the experimental data ( $y^{exp}$ ) and the predicted values obtained from the DNN model ( $y^m$ ). The likelihood function can be defined in different ways, and one option is the sum of squares function, as shown in Figure 4.11. In this equation, the residuals are between  $y^{exp}$  and  $y^m$ , where  $ny$  is the number of system outputs.

The prior density  $\mathbf{g}_\theta(\eta|I)$  contains the probability distribution of  $\theta$  before observing  $\eta$ . The marginal posterior density of  $\theta_j$  ( $\mathbf{g}_\theta(\eta|D, I)$ ) is then obtained by combining the likelihood  $L(\eta|D)$  and the prior  $\mathbf{g}_\theta(\eta|I)$ .

The optimal value of the weights and biases is  $\theta$  and is defined by the expected value of  $\mathbf{g}_\theta(\eta|D, I)$ , and the associated covariance matrix is also given by  $U_{\theta\theta}$  in Figure 4.11.

It should be noted that the solution for multiple integrations is not always possible to find analytically. In such cases, several methods can be used to solve the integrals, including approximations, Monte Carlo, and resampling. The MCMC method, proposed by Haario et al. (2006b), is particularly suitable for solving the inference problem of dynamic models, as suggested by Strawderman e Gamerman (2006). Therefore, the MCMC method using the adaptive metropolis algorithm is employed to solve the integrals.

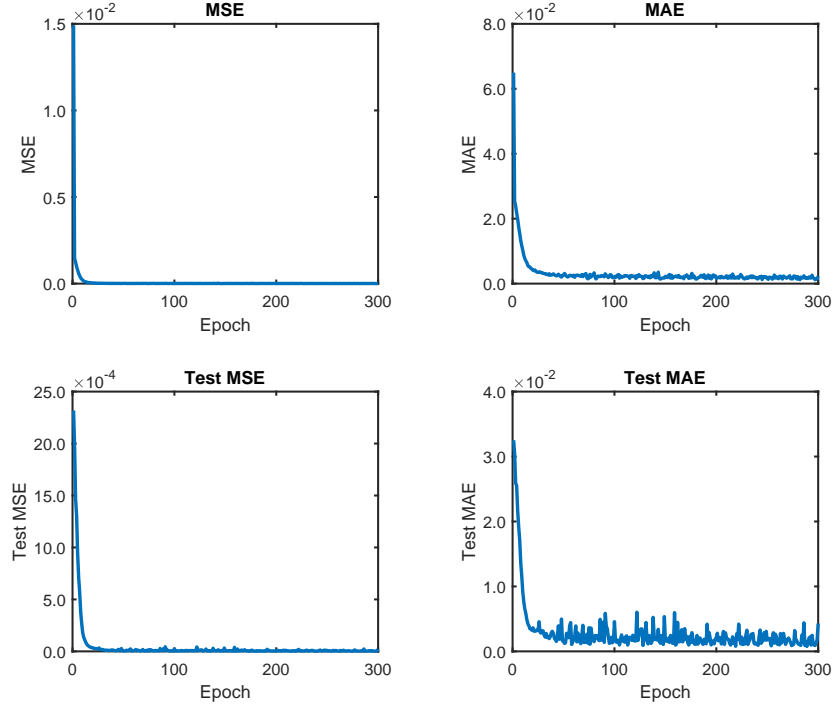


Figure 4.6: Intake pressure DNN MAE and MSE training and test values.

Section 4.3.3 shows the first training step performed using the ADAM (KINGMA; BA, 2017) algorithm by utilizing the train and test datasets and the parameters provided by Hyperband (LI et al., 2016). Géron (2019) discusses optimization algorithms in terms of speed and solution accuracy, among other things. However, this thesis seeks to validate other elements aimed at learning. Then, the ADAM algorithm was established as a standard for the preliminary step to obtaining a good starting point for the subsequent MCMC algorithm. Insufficiently optimal initial conditions may require many samples, potentially hindering chain convergence during the MCMC simulation. Upon completion of the simulation, a Markov chain is obtained for each neural network parameter. These chain values can be considered a parameter PDF component. Consequently, these sampled parameter values enable the propagation of neural network uncertainty for prediction purposes.

This first training provides a possible initial neural network to be evaluated before performing the Markov chain simulations. Also, these initial guess ( $\eta_0$ ) is used in the MCMC simulation to normalize all DNN weights and biases. This approach ensures that the MCMC algorithm will sort parameters in the same order. So the values drawn by the MCMC will be considered as  $\eta^* = \eta/\eta_0$  in which  $\eta^*$  is the normalized weights and bias,  $\eta_0$  is the initial guess and  $\eta$  is the real value. On the other hand, it is necessary to validate the initial guess to ensure that the proposed DNN can provide a good approximation of the system behavior. The first validation is presented in Figure 4.12, which shows the relations between the measured values and the neural network-predicted values.

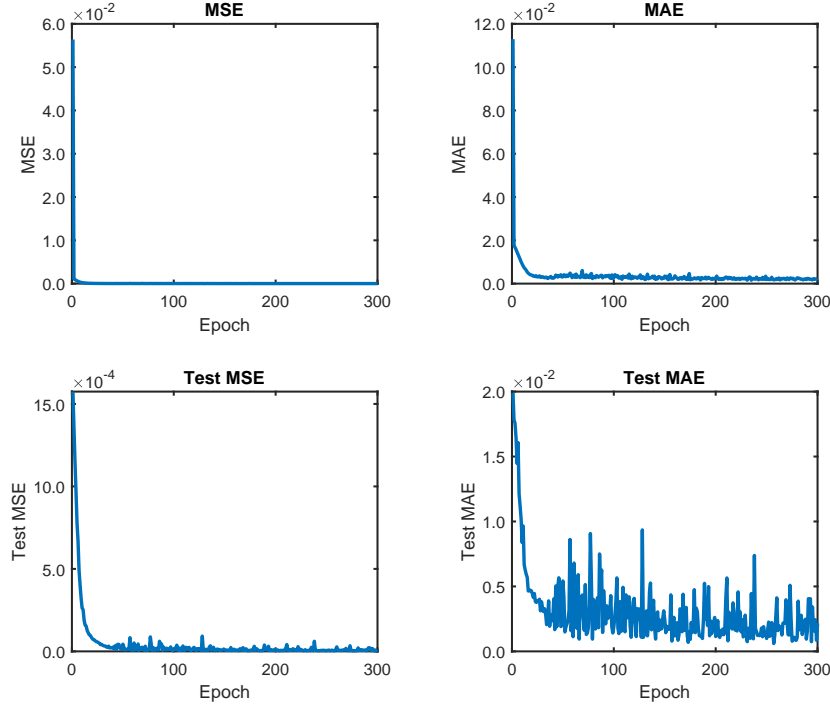


Figure 4.7: Production flow DNN MAE and MSE training and test values.

After the validation, the number of chain samples is set, and the MCMC algorithm picks samples until they reach the required PDF size. In this way, the models of  $p_c$ ,  $p_{in}$ , and  $q_c$  are sampled "m" times to obtain the desired PDF parameters.

The chain parameters provide information about the convergence of the MCMC process, particularly in verifying the chain's randomness. Figures 4.13 to 4.15 offer detailed information about the parameters of the PDF of the DNN weights. Graphs (A), (B), (D), and (F) present the marginalization of three randomly selected DNN parameters, allowing us to observe the correlation among these parameters during the chain-building process. In the 2D marginalization plots (Figures 4.13 - 4.15, panels B, D, and F), we can analyze the correlation matrix between the displayed parameters, explicitly referring to Equations 4 to 6. It's worth noting that due to the high-dimensional nature of the DNN model, the parameters are expected to exhibit some level of dependence or correlation without significantly impacting the model's predictive capacity.



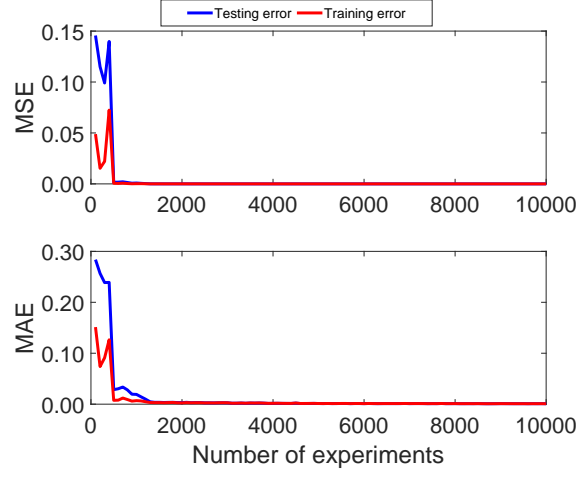


Figure 4.8: MAE and MSE experiment varying the amount of data for choke valve pressure.

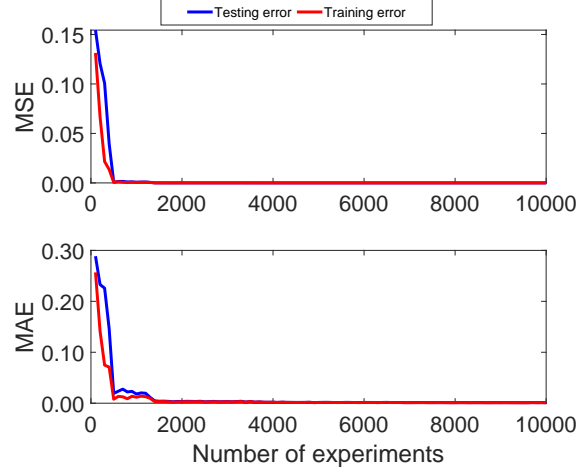


Figure 4.9: MAE and MSE experiment varying the amount of data for intake pressure.

$$\text{corr}_{P_c}\{2807, 1159, 1861\} = \begin{pmatrix} 1.0000 & 0.0600 & -0.0463 \\ 0.0600 & 1.0000 & 0.9920 \\ -0.0463 & 0.9920 & 1.0000 \end{pmatrix} \quad (4.4)$$

$$\text{corr}_{P_{in}}\{2829, 385, 1829\} = \begin{pmatrix} 1.0000 & 0.3865 & 0.2041 \\ 0.3865 & 1.0000 & 0.8889 \\ 0.2041 & 0.8889 & 1.0000 \end{pmatrix} \quad (4.5)$$

$$\text{corr}_{q_m}\{2829, 385, 1829\} = \begin{pmatrix} 1.0000 & -0.6598 & 0.0691 \\ -0.6598 & 1.0000 & -0.3875 \\ 0.0691 & -0.3875 & 1.0000 \end{pmatrix} \quad (4.6)$$

In the specific context presented, achieving convergence to a Gaussian distribution

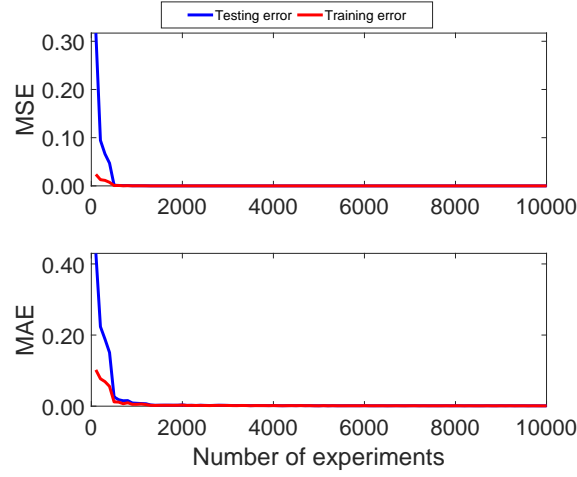


Figure 4.10: MAE and MSE experiment varying the amount of data for production flow.

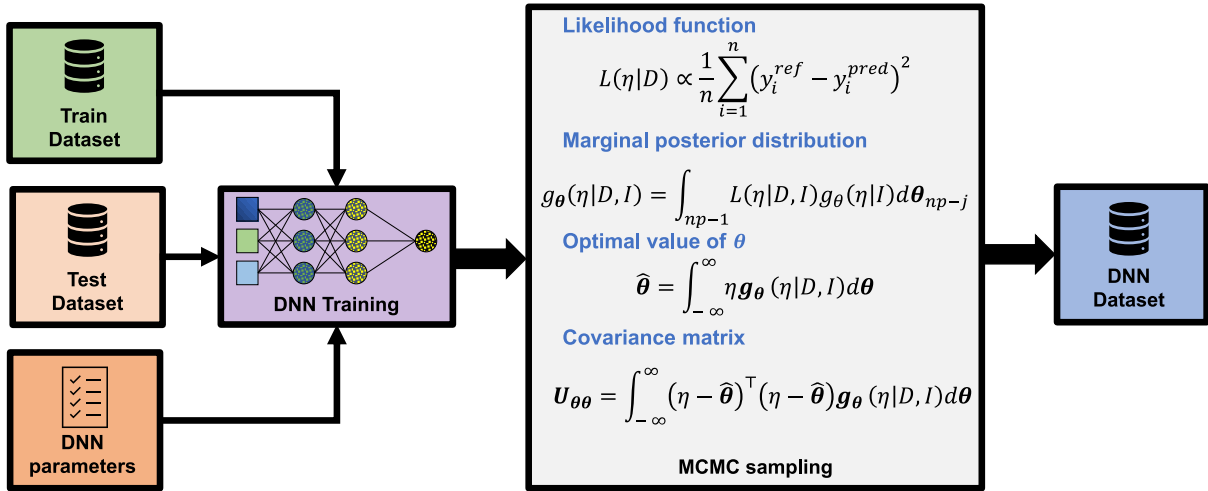


Figure 4.11: Monte Carlo training method.

for the parameters is unnecessary, as the MCMC solving process does not impose this assumption. However, Figures 4.13 - 4.15 (C) and (E) demonstrate that the mean and variance of the chain converge to Gaussian distributions. Besides, it is essential to assess the convergence of the chain. This analysis is accomplished using the Geweke parameter displayed in (G), which evaluates the relationship between the first 10% of the chain and the 50% of the final samples (BROOKS; ROBERTS, 1998). Geweke values close to one indicate chain convergence, and the histogram in (G) reveals that the parameter chains exhibit Geweke values relative to one. Table 4.3 provides an overview of the minimum, maximum, and values greater than 0.99, 0.999, and 0.9999. By examining the values in Table 4.3, it is possible to observe that most parameters have Geweke values exceeding 0.99, indicating satisfactory convergence of the networks.

To further understand the behavior of the chain, Figures 4.13-4.15 (H), (J), and (L) show the burn-in samples used to obtain an initial estimate of the PDF of the parameters

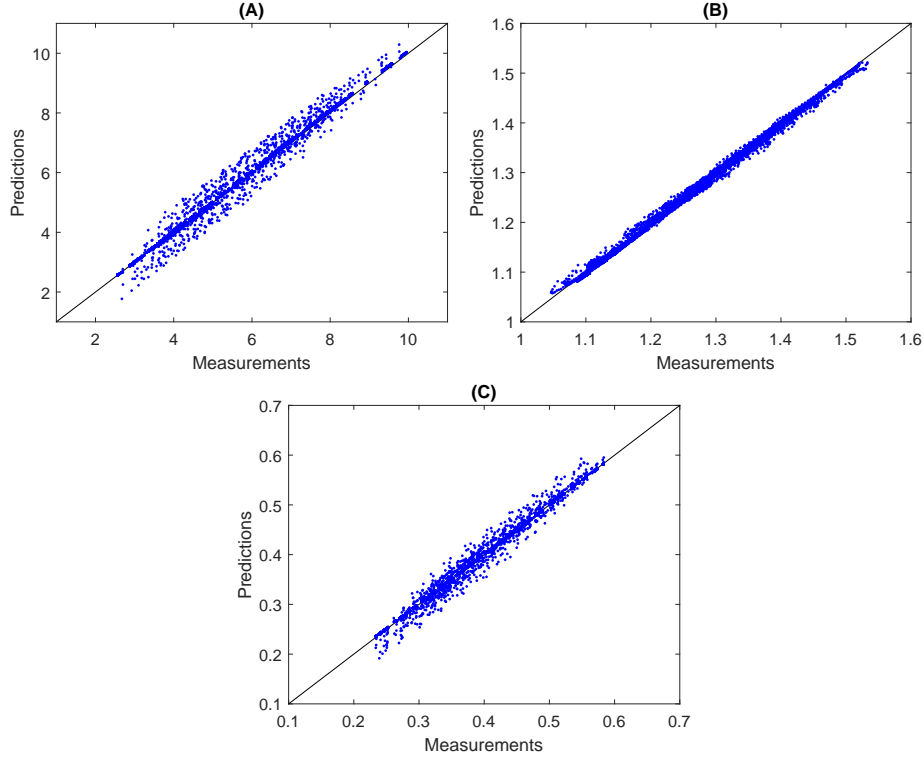


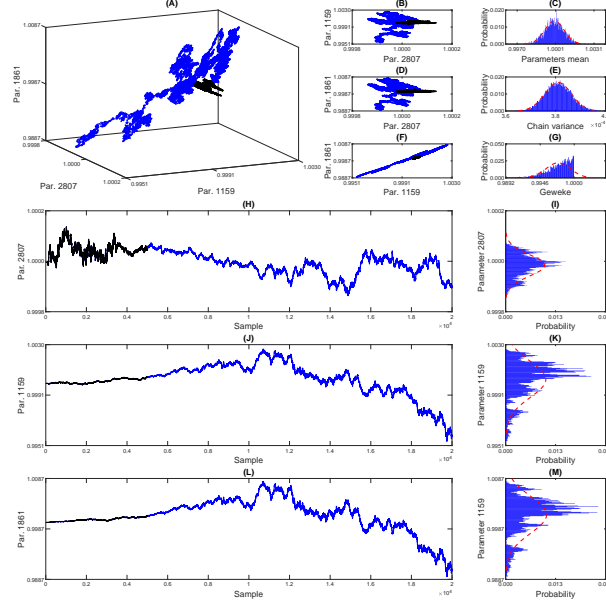
Figure 4.12: Parity chart between the measured and DNN predicted values. In (A) is the intake pressure, in (B) is the choke pressure, and in (C) is the flow rate.

Table 4.3: Geweke statistics.

Chain	$P_{in}^*$	$P_c$	$q_c$
Min/ADM	0.(9) <sub>3</sub>	0.(9) <sub>2</sub>	0.(9) <sub>3</sub>
Max/ADM	0.(9) <sub>8</sub>	0.(9) <sub>8</sub>	0.(9) <sub>8</sub>
Percentage higher than 0.(9) <sub>2</sub> /%	100.00	99.82	100.00
Percentage higher than 0.(9) <sub>3</sub> /%	99.97	25.63	100.00
Percentage higher than 0.(9) <sub>4</sub> /%	30.98	3.01	30.98

\* The notation 0.(9)<sub>n</sub> means 9 repeated  $n$  times.

and are discarded at the end of the process, as these initial samples may be far from the stationary behavior of the chain (section 4.8 of Strawderman e Gamerman (2006)). Visually, we can observe significant variations in the values in these figures. However, since the parameters are normalized concerning unity, the variations proportional to the initial value are less than 0.07% of the initial value. This demonstrates the stability of the region where the chain is being built and suggests it is a viable candidate for a minimum point.

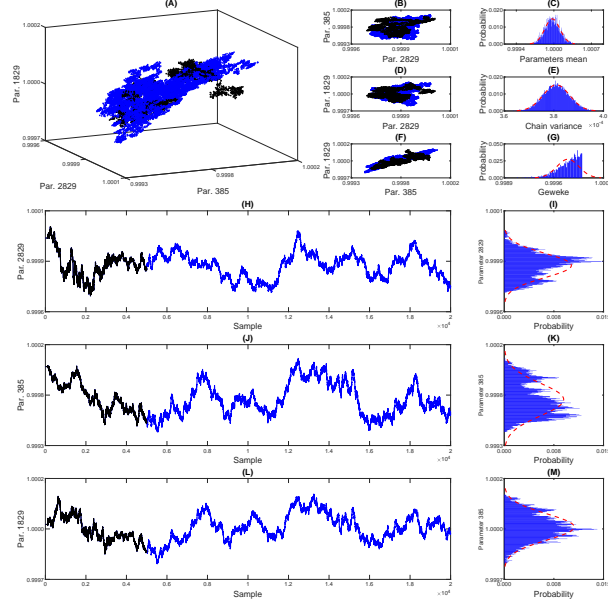
Figure 4.13: Chain analysis of  $P_c$ .

#### 4.3.5 Propagation and experimental validation

The final step of the proposed methodology is to validate the model with experimental data. In this work, the experimental data were obtained in the ESP system of Figure 3.4. The signals presented in Figure 4.16 were inserted into the experimental system. After data collection, the values are entered into the DNN models to obtain the simulated responses. With the simulated results, it is possible to compare the dynamic responses and validate the DNN model against experimental data.

Figure 4.16 shows the variables measured in the ESP system installed at CTAI/UFBA. The ESP system operated continuously for 70h, and input variables were choke valve opening level ( $Z_c$ ), ESP motor frequency ( $f$ ), and reservoir valve opening level ( $Z_r$ ). The ESP system was excited with four different ESP motor frequencies (55 Hz, 50 Hz, 45 Hz, and 35 Hz), three different choke valve opening levels (100 %, 95 %, and 80 %), and a reservoir valve opening level was kept constant (100 %) throughout the experimental test. The ESP motor frequency, Figure 4.16 (A), is a fast action/reaction input directly to the variable speed drive (VSD). The choke valve (Figure 4.16 (B)) is considered a fast-acting (electrically activated) and slow-reacting (mechanical parts activated) inlets. The reservoir pressure (Figure 4.16 (C)) and the manifold pressure (Figure 4.16 (D)) are variables that suffer interference from the frequency and the opening of the choke valve but are considered as exogenous variables for the models built here.

Figures 4.17, 4.18, and 4.19 illustrate the system's dynamic behavior through sets of experimental tests, each displaying the respective range of signals. Notably, the DNN's most probable values closely align with the experimental data in each graph. These figures also showcase the prediction uncertainty limits of the DNNs in black, alongside the experimental uncertainties represented in blue. The DNNs' prediction uncertainty is

Figure 4.14: Chain analysis of  $P_{in}$ .

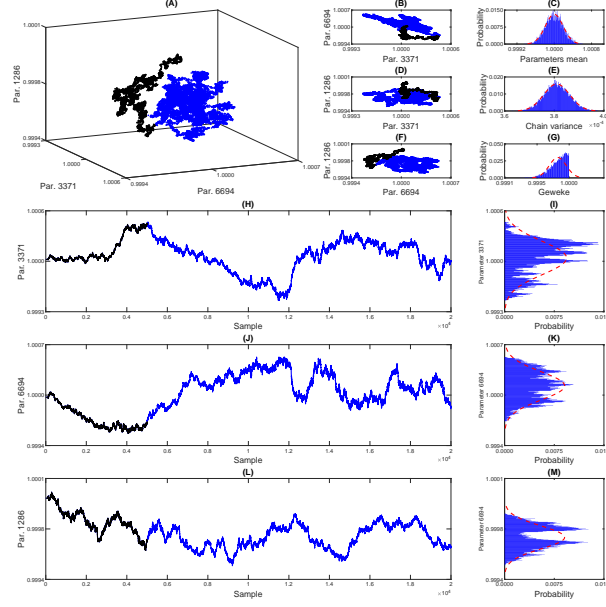
obtained by propagating the uncertainty from the PDF of the weights and biases using a Monte Carlo simulation. Meanwhile, the experimental data uncertainty is assessed using GUM methods, assuming the signal is steady. To ensure comparability with the uncertainties of the DNN models, the analysis by Costa et al. (2021) is utilized.

It's important to highlight that the uncertainties of the experimental data are depicted only at points where the system can be considered in a steady state (as indicated by Costa et al. (2021) analysis). Consequently, based on the comprehensive uncertainty analysis presented, we conclude that the DNN models are successfully validated, as the coverage regions of the DNN predictions overlap with the uncertainties of the experimental data. This outcome further reinforces the reliability and accuracy of the DNN models in predicting the system's behavior.

#### 4.4 CONCLUSIONS

This chapter introduces an innovative approach to deep-learning modeling of ESP systems, which incorporates uncertainty assessment. The primary challenge in addressing this approach is the need for extensive data to train and validate DNN models for ESP. The study utilizes synthetic data generated by an experimentally validated nonlinear simulator capable of capturing diverse process dynamics to overcome the challenge of data dependency. Bayesian inference is then employed to assess the uncertainty of the DNN models, with MCMC simulations used to generate the PDF of DNN parameters. Subsequently, the performance of the identified DNN models is evaluated using experimental data in real-case scenarios.

The results demonstrate that a NARX model using a DNN with fully connected layers accurately represents the behavior of the ESP system. The uncertainty assessment con-

Figure 4.15: Chain analysis of  $Q_c$ .

firmly that the DNN models statistically align with the nonlinear model and experimental data, with overlapping coverage regions.

As a result, the NARX DNN models, validated using experimental data, successfully predict parameters of an ESP system, including choke pressure ( $p_c$ ), intake pressure ( $p_{in}$ ), and production flow ( $q_c$ ). These findings contribute valuable insights to drive the development of artificial intelligence applications in the ESP field. Consequently, it provides the basis for exploration in subsequent stages of development, including digital twins, soft sensors, control, optimization, and production monitoring.

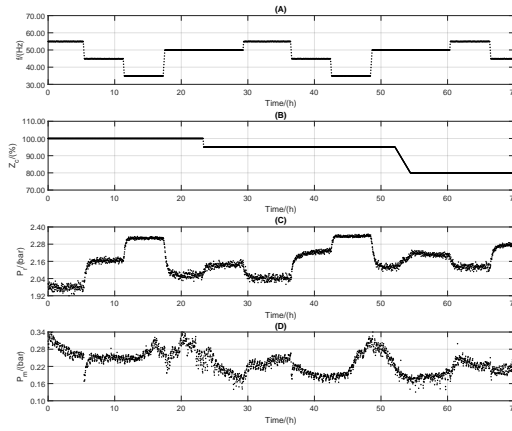


Figure 4.16: Input variables: (A) ESP motor frequency ( $f$ ), (B) valve opening level ( $Z_c$ ), (C) reservoir pressure ( $P_r$ ), and manifold pressure ( $P_m$ ).

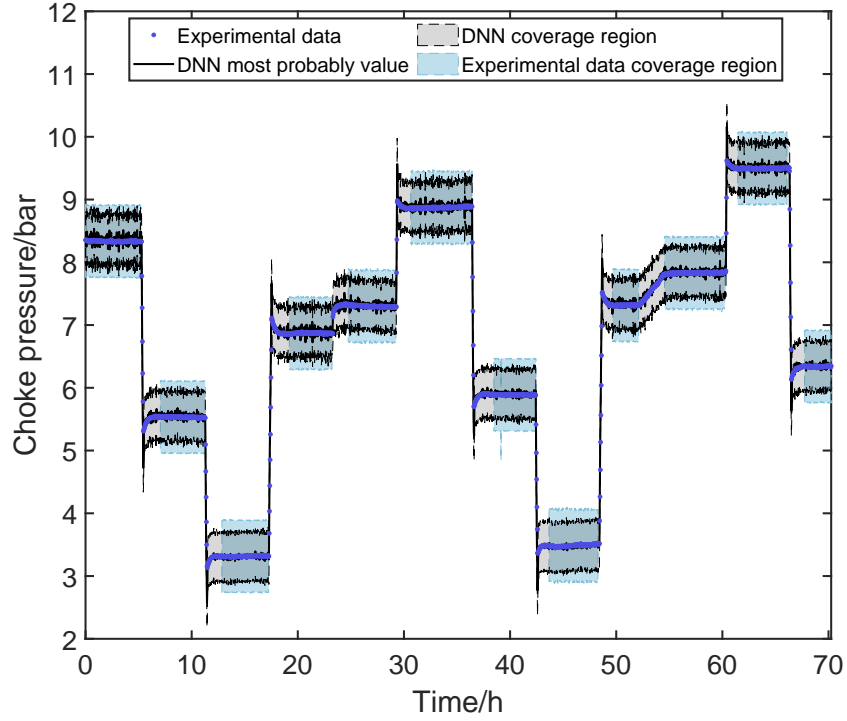
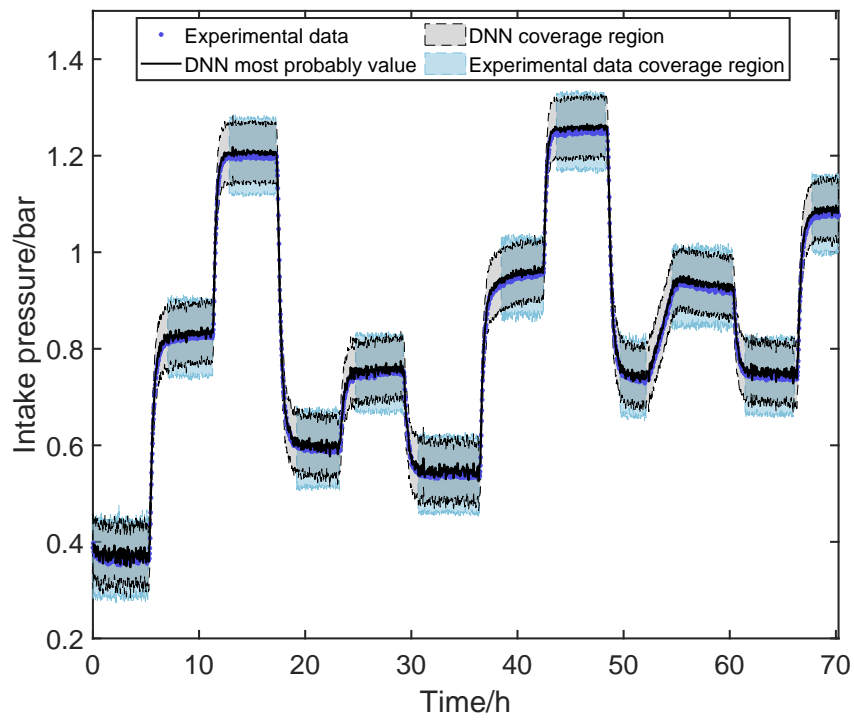
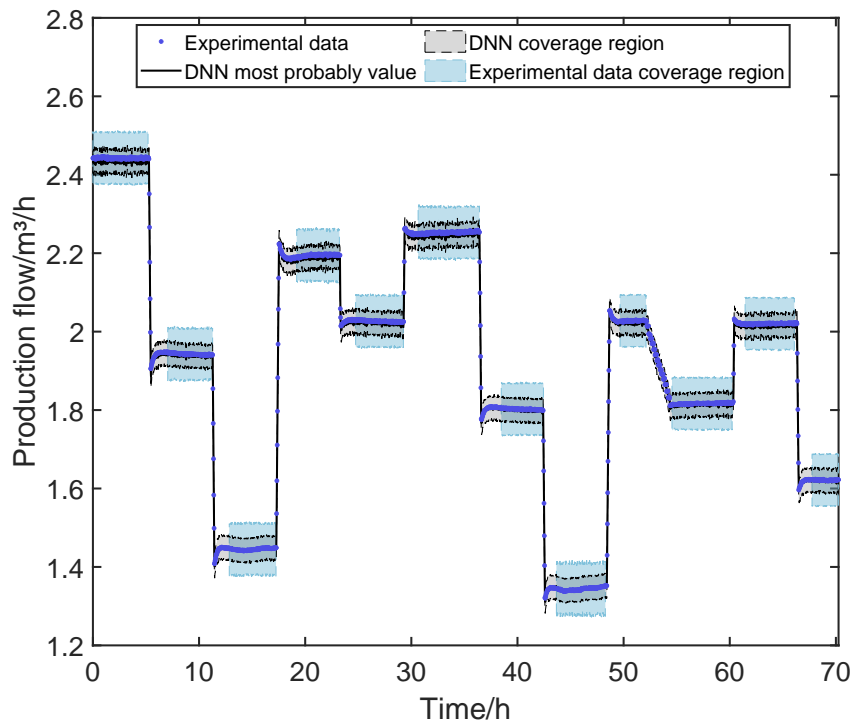


Figure 4.17: Experimental validation of  $P_c$ .

Figure 4.18: Experimental validation of  $P_{in}$ .Figure 4.19: Experimental validation of  $Q_c$ .



## CHAPTER 5

# ADAPTIVE DIGITAL TWIN FOR PRESSURE SWING ADSORPTION SYSTEMS\*

\*Published on: **Costa, E. A.**; Rebello, C. M.; Schnitman, L.; Loureiro, J. M.; Ribeiro, A. M.; & Nogueira, I. B. R. (2024). Adaptive digital twin for pressure swing adsorption systems: Integrating a novel feedback tracking system, online learning and uncertainty assessment for enhanced performance. *Engineering Applications of Artificial Intelligence*, 127, 107364.

DOI: <https://doi.org/10.1016/J.ENGAPPAI.2023.107364>.

### 5.1 INTRODUCTION

This chapter presents a novel approach to digitizing and modeling pressure swing adsorption (PSA) processes using an uncertainty-aware digital twin. PSA modeling presents unique challenges due to its complex and cyclic behavior, which lacks a steady state. By contributing to the literature on periodic systems, we provide valuable insights into the potential applications of artificial intelligence and digital twins beyond the field of cyclic processes. Also, in this Chapter, the four hypotheses presented in section 1.1 are evaluated in a broader context of building the digital twin. Our proposed methodology can enhance the understanding and optimization of complex systems across various industries and applications. The proposed digital twin is uncertainty-aware and reliable, continuously updating itself through online learning (OL) and utilizing a novel feedback tracker to represent the PSA system accurately. This new, robust, adaptable methodology supports optimal PSA operation and facilitates decision-making for enhanced process operation. The results demonstrate that the proposed approach yields a reliable digital twin for the PSA unit, capable of tracking the process's complex dynamics and adapting to changes, including adsorbent degradation, which is a significant challenge in PSA operations. Overall, this work highlights the potential of advanced technologies, such as digital twins and artificial intelligence, to improve performance and efficiency in process engineering. This work contributes to the ongoing efforts to optimize industrial processes and support sustainable development by providing a reliable and adaptable methodology for digitizing PSA processes.

### 5.2 METHODOLOGY AND RESULTS

This section outlines the methodology proposed by this Chapter. The focus is to develop an artificial intelligence-based digital twin (DT) capable of representing the PSA system and evaluating the prediction uncertainties of the DNN model. Additionally, the DT can update the DNN model using an OL approach using the proposed feedback tracker.

The methodology is divided into two primary steps, as illustrated in the Figure 5.1 flowchart. The first step involves the development of the DT, which includes training and uncertainty assessment of the DNN model. The second step encompasses the application phase, where the feedback tracker and DT online learning updates are carried out. The methodology flowchart can be found in Figure 5.1, and the following sections provide detailed information on the methodology and corresponding results.

**DT Development Step:** This step involves constructing and training the DNN model, followed by an uncertainty assessment. The DNN model is designed to accurately represent the PSA system's dynamics, considering factors such as pressure swings and adsorption processes. A comprehensive set of historical data is utilized to train the DNN, ensuring it can effectively predict the system's behavior. This data was generated by applying the design of experiments into a virtual plant previously validated at the laboratory. Subsequently, an uncertainty assessment is conducted to evaluate the model's reliability, accounting for potential inaccuracies in data and model structures.

**Application Step:** In this phase, the DT is integrated into the PSA virtual plant to monitor its performance and provide real-time feedback. A feedback tracker is implemented to compare the actual system behavior with the DNN model's predictions, identifying discrepancies and potential areas for improvement. Hence, performing feedback control at the DT representability of the system. The DT online learning update is then employed, allowing the model to adapt and refine its predictions based on the newly acquired data. This continuous updating process ensures that the DT remains accurate and reliable as the system evolves.

### 5.2.1 Pressure swing adsorption virtual plant

To identify the DNN models, a first-principle model was employed to characterize the dynamic behavior of the pressure swing adsorption unit. This phenomenological model served as a virtual plant and supplied the synthetic data for the DNN model identification, all within a software-in-the-loop approach. Furthermore, the model was used as a deployed environment where the DT was installed to track the virtual plant and monitor its performance.

The phenomenological model is described by mass, energy, and momentum balances, adsorption equilibrium and kinetics, and their respective boundary conditions. It was proposed in Regufe et al. (2015), Rebello et al. (2022b) and assumed ideal gas behavior throughout the column. Other assumptions include no mass, heat, or velocity gradients in the radial direction; axial dispersed plug flow; external mass and heat transfer resistances represented by the film model; internal mass transfer resistance depicted by the Linear Driving Force (LDF) model; absence of temperature gradients inside each particle since heat transfer in solid particles is much faster than in the gas phase; constant porosity along the bed; and local validity of the Ergun equation, meaning the momentum balance only considers pressure drop and velocity change terms. Experimentally validated the model (REGUFE et al., 2015), which assures the reliability of the results presented in this work.

In the presented case study, a PSA unit operates in five stages: co-current pressur-

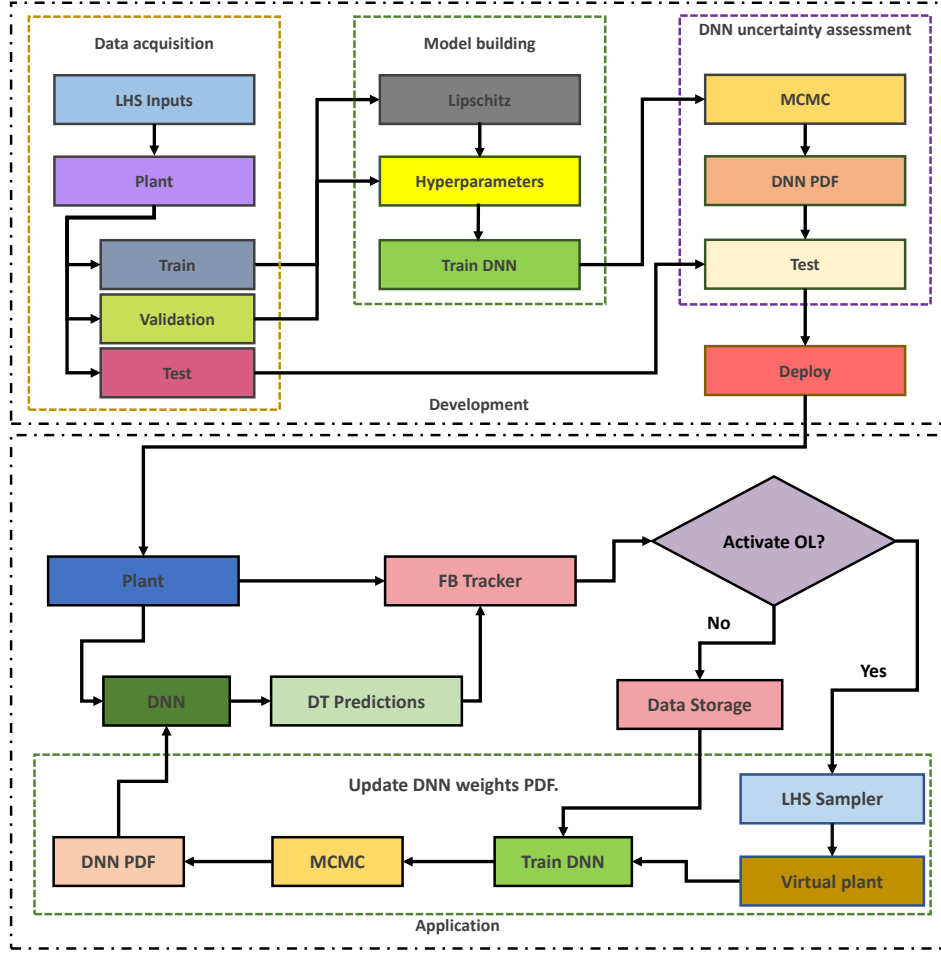


Figure 5.1: General methodology flowchart. The abbreviations are: LHS - Latin Hypercube Sampling, DNN - Deep neural network, MCMC - Markov Chain Monte Carlo, PDF - Probability Density Function, FB - Feedback Tracker, OL - Online Learning.

ization, feed, rinse, blowdown, and purge, as depicted in Figure 5.2. During the pressurization and blowdown stages, pressure modulation occurs, while the pressure remains constant in the other stages. Regufe et al. (2015) propose that this mode of operation seeks to generate a  $CO_2$ -enriched stream and a stream with an  $H_2/CO$  stoichiometric ratio ranging from 2.2 to 2.3, which is the ideal feed range for the Fischer-Tropsch process. Table 5.1 presents the operating limits for the PSA process studied here. These limits were used to develop the next steps of the proposed methodology.

This Chapter uses a model for a PSA unit that is articulated through a set of partial differential-algebraic equations. These equations reflect the unit's dynamics by incorporating mass, energy, and momentum balances alongside constitutive relationships that portray kinetic and dynamic equilibrium adsorption. The intricacies of the PSA dynamics revolve around a fixed-bed column, progressing in a series of steps. The following steps are to define the model utilized as a virtual plant under the assumptions delineated below:

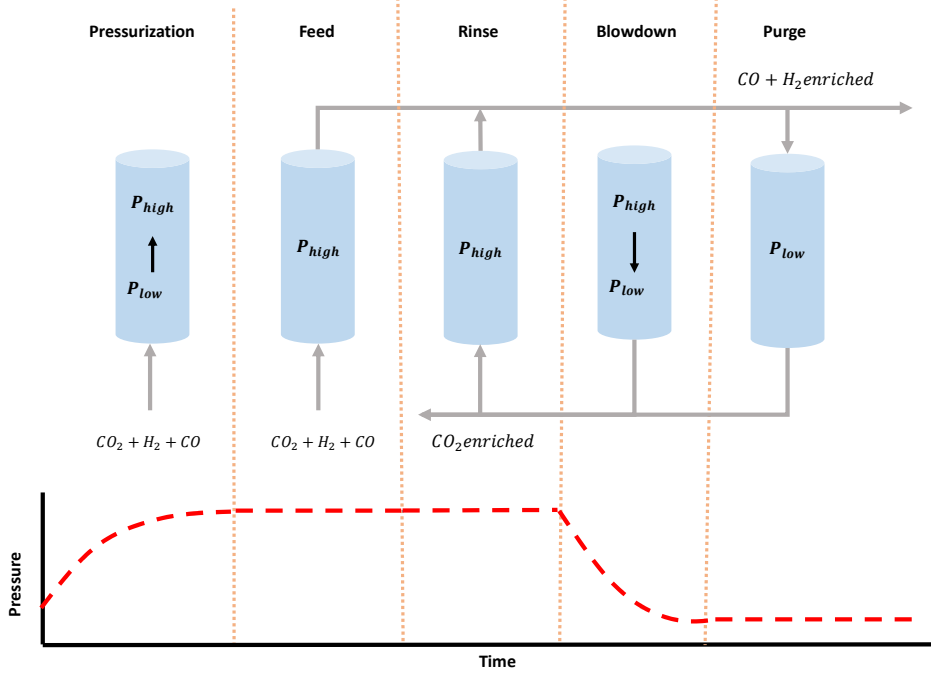


Figure 5.2: Pressure swing adsorption stages.

Table 5.1: Inputs variables limits.

Variable	min	max
$t_{feed}/s$	280.00	715.00
$t_{purge}/s$	80.00	115.00
$t_{rinse}/s$	187.00	265.00
$P_{high}/Pa$	$3.40 \times 10^5$	$5.00 \times 10^5$
$P_{low}/Pa$	$5.50 \times 10^4$	$1.20 \times 10^5$
$Q_{rinse}/SLPM$	0.20	0.60
$Q_{purge}/SLPM$	0.20	0.60
$T_{inlet}/K$	304.00	378.15

- The behavior of an ideal gas is consistent throughout the column.
- There are no mass, heat, or velocity gradients in the radial direction.
- The flow is characterized by axial dispersed plug flow.
- External mass and heat transfer resistances are outlined using the film model.
- The Linear Driving Force (LDF) model is used to describe internal mass transfer resistance.
- Considering that the heat transfer in solid particles is significantly faster than in the gas phase, there are no temperature gradients within each particle.

- Porosity remains consistent along the bed.
- The Ergun equation is applicable locally. This means in the momentum balance, only pressure drop and velocity change terms are considered.

The equation of the mass balance for each component in the gas phase is:

$$\frac{\partial}{\partial z} \left( \varepsilon D_{ax} C_{g,T} \frac{\partial y_i}{\partial z} \right) - \frac{\partial}{\partial z} (u_0 C_{g,i}) - \varepsilon \frac{\partial C_{g,i}}{\partial t} - (1 - \varepsilon) a_p k_f (C_{g,i} - C_{s,i}) = 0 \quad (5.1)$$

wherein  $z$  stands for the axial position and  $t$  denotes time. The bed porosity is represented by  $\varepsilon$ , while the superficial velocity is given by  $u_0$ . Both the total and  $i^{th}$  component gas phase concentrations are represented by  $C_{g,T}$  and  $C_{g,i}$ , respectively. The  $i^{th}$  component molar fraction is expressed as  $y_i$ , and  $C_{s,i}$  signifies the concentration of the  $i^{th}$  component at the solid interface. The coefficient of mass axial dispersion is presented as  $D_{ax}$ , the film mass transfer coefficient is noted as  $k_f$ , and  $a_p$  refers to the external specific area of the particle.

The momentum balance, represented by the Ergun equation, articulates the pressure drop and velocity change as:

$$-\frac{\partial P}{\partial z} = \frac{150\mu(1-\varepsilon)^2}{\varepsilon^3 d_p^2} u_0 + \frac{1.75(1-\varepsilon)\rho}{\varepsilon^3 d_p} |u_0| u_0 \quad (5.2)$$

In this equation,  $P$  denotes the total pressure,  $\mu$  the gas viscosity,  $\rho$  the gas density, and  $d_p$  the particle diameter.

The energy balance within the gas phase can be expressed as:

$$\begin{aligned} \frac{\partial}{\partial z} \left( \lambda \frac{\partial T_g}{\partial z} \right) - u_0 C_{g,T} C_p \frac{\partial T_g}{\partial z} + \varepsilon R_g T_g \frac{\partial C_{g,T}}{\partial t} - (1 - \varepsilon) a_p h_f (T_g - T_p) - \frac{4h_w}{d_{wi}} (T_g - T_w) \\ - \varepsilon C_{g,T} C_V \frac{\partial T_g}{\partial t} = 0 \end{aligned} \quad (5.3)$$

Here,  $T_g$ ,  $T_p$ , and  $T_w$  represent the temperatures of the gas, particle, and wall, respectively;  $C_v$  and  $C_p$  signify the gas molar specific heats at constant volume and pressure;  $R_g$  is the ideal gas constant;  $d_{wi}$  is the wall internal diameter, and  $\lambda$  represents the heat axial dispersion coefficient. The film heat transfer coefficients between the gas phase, particle, and wall are denoted as  $h_f$  and  $h_w$ , respectively.

The mass transfer rates within the solid phase are illustrated by the Linear Driving Force model (LDF). The mass balance in the solid phase is represented as:

$$\frac{\partial \bar{q}_i}{\partial t} = \frac{15D_{p,i}}{R_p^2} (q_i^* - \bar{q}_i) \quad (5.4)$$

In this equation,  $D_{p,i}$  is the pore diffusivity,  $R_p$  is the particle radius,  $\bar{q}_i$  is the particle averaged adsorbed concentration, and  $q_i^*$  is the adsorbed concentration in equilibrium with  $C_{s,i}$ , which is calculated using the multicomponent extension of the Langmuir isotherm:

$$q_i^* = q_{m,i} \frac{K_i P_i}{1 + \sum_{j=1}^n K_j P_j} \quad (5)$$

The isotherm parameters were derived from the work of Regufe et al. (2015) and are delineated in Table 5.2.

Table 5.2: Langmuir fitting parameters of the model for CO<sub>2</sub>, CO and H<sub>2</sub> adsorption equilibrium on MIL-125(Ti)<sub>NH2</sub>.

Species	$q_m$ (mol·kg <sup>-1</sup> )	$K_i^0$ (bar <sup>-1</sup> )	$(-\Delta H)$ (kJ·mol <sup>-1</sup> )
CO <sub>2</sub>	8.509	$0.5782 \times 10^{-5}$	21.9
CO	5.229	$3.2700 \times 10^{-4}$	11.9
H <sub>2</sub>	0.294	$9.3900 \times 10^{-7}$	35.3

The fluxes at the particle surface bind Equations 5.1 and 5.4, as expressed by:

$$\frac{(1 - \varepsilon) a_p k_f}{\rho_b} (C_{g,i} - C_{s,i}) = \frac{15 D_{p,i}}{R_p^2} (q_i^* - \bar{q}_i) \quad (6)$$

Here,  $\rho_b$  signifies the bulk density of the bed.

In the absence of temperature gradients within an adsorbent particle, the adsorbed phase and solid phase energy balances are characterized by:

$$(1 - \varepsilon) \left[ \rho_p \sum_{i=1}^n \bar{q}_i C_{v,ads,i} + \rho_p \hat{C}_{ps} \right] \frac{\partial T_p}{\partial t} = \rho_b \sum_{i=1}^n (-\Delta H_{ads})_i \frac{\partial \bar{q}_i}{\partial t} + (1 - \varepsilon) a_p h_f (T_g - T_p) \quad (7)$$

In this equation,  $\hat{C}_{ps}$  represents the solid specific heat per mass unit, and  $(-\Delta H_{ads})_i$  is the heat of adsorption of component  $i$ .

An energy balance to the column wall is also considered, where the energy exchanges with the gas phase inside the column and the external environment are illustrated:

$$\rho_w \hat{C}_{p,w} \frac{\partial T_w}{\partial t} = \alpha_w h_w (T_g - T_w) - \alpha_w l U (T_w - T_\infty) \quad (8)$$

Here,  $T_\infty$  denotes the external temperature,  $\rho_w$  is the wall density,  $\hat{C}_{p,w}$  is the wall-specific heat per mass unit,  $U$  is the overall heat transfer coefficient, and  $\alpha_w$  and  $\alpha_{wl}$  are defined by equations (9) and (10):

$$\alpha_w = \frac{d_{wl}}{e(d_{wl} + e)} \quad (9)$$

$$\alpha_{wl} = \frac{2}{(d_{wl} + e) \ln \left( \frac{d_{wl} + 2e}{d_{wl}} \right)} \quad (10)$$

The term  $e$  indicates the wall thickness.

Simulating the PSA process requires adopting appropriate boundary conditions for each step, which are detailed in Table 5.3. Also, the physical parameters are given in Table 5.4

Table 5.3: Boundary conditions.

Pressurization with feed	
$z=0$ , inlet	$z=L$
$u_{0,\text{inlet}} C_{\text{inlet},i} = u_0 C_{g,i} - \varepsilon D_{ax} C_{g,T} \frac{\partial y_i}{\partial z}$	$\frac{\partial C_{g,i}}{\partial z} = 0$
$P = P_{\text{inlet}}$	$u_0 = 0$
$u_{0,\text{inlet}} C_{\text{inlet},T} C_p T_{\text{inlet}} = u_0 C_{g,T} C_p T_g - \lambda \frac{\partial T_g}{\partial z}$	$\frac{\partial T_g}{\partial z} = 0$
Feed	
$u_{0,\text{inlet}} C_{\text{inlet},i} = u_0 C_{g,i} - \varepsilon D_{ax} C_{g,T} \frac{\partial y_i}{\partial z}$	$\frac{\partial C_{g,i}}{\partial z} = 0$
$u_{0,\text{inlet}} C_{\text{inlet},T} = u_0 C_{g,T}$	$P = P_{\text{outlet}}$
$u_{0,\text{inlet}} C_{\text{inlet},T} C_p T_{\text{inlet}} = u_0 C_{g,T} C_p T_g - \lambda \frac{\partial T_g}{\partial z}$	$\frac{\partial T_g}{\partial z} = 0$
Counter-current blowdown	
$\frac{\partial C_{g,i}}{\partial z} = 0$	$\frac{\partial C_{g,i}}{\partial z} = 0$
$P = P_{\text{outlet}}$	$u_0 = 0$
$\frac{\partial T_g}{\partial z} = 0$	$\frac{\partial T_g}{\partial z} = 0$
Purge	
$\frac{\partial C_{g,i}}{\partial z} = 0$	$u_{0,\text{inlet}} C_{\text{inlet},i} = u_0 C_{g,i} - \varepsilon D_{ax} C_{g,T} \frac{\partial y_i}{\partial z}$
$P = P_{\text{outlet}}$	$u_{0,\text{inlet}} C_{\text{inlet},T} = u_0 C_{g,T}$
$\frac{\partial T_g}{\partial z} = 0$	$u_{0,\text{inlet}} C_{\text{inlet},T} C_p T_{\text{inlet}} = u_0 C_{g,T} C_p T_g - \lambda \frac{\partial T_g}{\partial z}$

### 5.2.2 Digital twining - Development steps

The development methodology adopts a comprehensive approach to create DNNs models that accurately represent the system and assess the model's prediction uncertainties using Bayesian inference. The initial stage involves data acquisition, where training, testing, and validation datasets are generated to provide a solid foundation for model development, as seen in Figure 5.1. Various uncorrelated inputs are employed to create the datasets to stimulate the system and acquire the necessary system output variables. The Latin Hypercube Sampling (LHS) algorithm generates the input set, ensuring sample independence and adequate input space coverage. This diverse input set triggers the system, generating a corresponding variable output response.

To build the digital twin, this article used synthetic data generated from the complete mathematical model of the PSA system. This approach becomes interesting with a validated model, as it allows obtaining considerable training, testing, and validation data. Likewise, this approach implies little dependence on the specialist knowledge of those who operate the plant since experimental data from real systems imply specific operating conditions limited to a region. Also, the use of synthetic data allows the entire admissible range for the variables, as well as the respective combinations, to be tested and incorporated into the data set used for the construction of the digital twin.

Figure 5.3 illustrates the dimensionless distribution of experimental points generated using the LHS technique. The scatter plot displays a well-dispersed set of 50000 points within the search space's boundaries, highlighting the efficiency of the LHS method in ensuring comprehensive coverage of the input space, defined in Table 5.1. Each axis

Table 5.4: Physical parameters and values

Parameters	Value
<b>Properties of fluids and adsorbents</b>	
Pore diffusivity, $D_p$ ( $\text{m}^2 \cdot \text{s}^{-1}$ )	CO2: $7.7 \times 10^{-7}$ , CO: $9.8 \times 10^{-7}$ , H2: $2.5 \times 10^{-6}$
Mass axial dispersion coefficient, $D_{\text{ax}}$ ( $\text{m}^2 \cdot \text{s}^{-1}$ )	$6.9 \times 10^{-5}$
Film mass transfer coefficient, $k_f$ ( $\text{m} \cdot \text{s}^{-1}$ )	$2.0 \times 10^{-2}$
Heat transfer coefficient between gas and particle, $h_f$ ( $\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$ )	128
Overall heat transfer coefficient, $U$ ( $\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$ )	37
Heat axial dispersion coefficient, $\lambda$ ( $\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$ )	0.6
Heat transfer coefficient between gas phase and wall, $h_w$ ( $\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$ )	50
Fluid viscosity, $\mu$ ( $\text{Pa} \cdot \text{s}$ )	$1.7 \times 10^{-5}$
Heat capacity of the mixture at constant pressure, $c_p$ ( $\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ )	31
Heat capacity of the mixture at constant volume, $c_v$ ( $\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ )	23
<b>Characteristics of granulate MIL-125 (Ti) NH2</b>	
Bed length, $L$ (m)	0.323
Bed diameter, $d_{\text{in}}$ (m)	0.021
Bed porosity, $\varepsilon$	0.35
Mass of adsorbent, $m_a$ (kg)	0.0402
Particle radius, $r_p$ (mm)	2.00
Solid density, $\rho_p$ ( $\text{kg} \cdot \text{m}^{-3}$ )	1450
Apparent particle density, $\rho_{\text{ap}}$ ( $\text{kg} \cdot \text{m}^{-3}$ )	550
<b>Feeding conditions</b>	
Feed flow rate (SLPM)	0.43
Feed composition (%)	CO2: 30%, CO: 22%, H2: 48%

represents a different input variable, and the points are uniformly spread across the entire range of possible values for each variable.

This even distribution of data points is crucial for synthetic data generation and subsequent AI model training, as it enables the model to learn the underlying relationships



between the input variables and system output more effectively. The AI model can be trained on diverse input-output pairs by ensuring the data covers the entire search space, promoting generalization and improving its predictive capabilities.

Hence, Figure 5.3 visualizes the LHS-generated experimental points, emphasizing the importance of well-distributed data for AI models' successful training and performance in pressure swing adsorption systems.

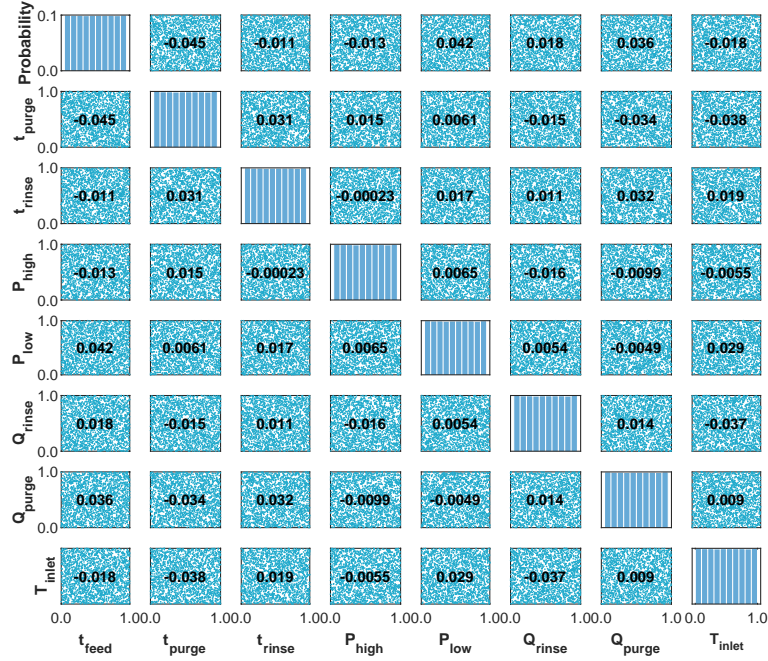


Figure 5.3: Correlation map of inputs.

Once the datasets are prepared, the next crucial step is identifying the appropriate predictor structure for the AI model. Various predictor structures exist, such as nonlinear autoregressive with exogenous inputs (NARX), autoregressive network (ARN), nonlinear output error (NOE), and others. In this study, the NARX structure was chosen due to its simplicity and widespread recommendation in literature as the most suitable structure for prediction purposes (REBELLO et al., 2022a; NOGUEIRA et al., 2020b; MARTINS et al., 2021).

The NARX model is particularly advantageous for capturing complex, nonlinear relationships between input and output variables in dynamic systems. One of the key aspects of the NARX structure is its embedding dimensions, which need to be estimated during the model development process. These embedding dimensions comprise the number of delays for both the output and input variables. Determining a good value for these delays is essential, as it directly affects the model's performance in capturing the underlying system dynamics and making accurate predictions.

Choosing the NARX structure for the AI model ensures that the predictor can handle the complexities associated with pressure swing adsorption systems while maintaining a

relatively simple and computationally efficient model architecture. Properly estimating the embedding dimensions further enhances the model's predictive capabilities, ultimately leading to a more reliable and effective AI-driven solution for monitoring.

In addition to its simplicity and suitability for prediction purposes (NOGUEIRA et al., 2020a, 2022), the NARX model can also benefit from using the same method applied in Subsection 2.2.3 to identify the predictor's embedding dimensions. Thus, Figure 5.4 presents Lipschitz index results, and it was possible to define that the minimum value for each delay input and output is two. However, this work adopted 5 delays for the inputs to ensure the model has enough information. The justification for this came from the fact that there are no significant changes in the Lipschitz slope for the inputs, Figure 5.4. The measurement of input variables is typically readily available in the process. Therefore, these variables can be judiciously leveraged to enhance the model's predictive capabilities.

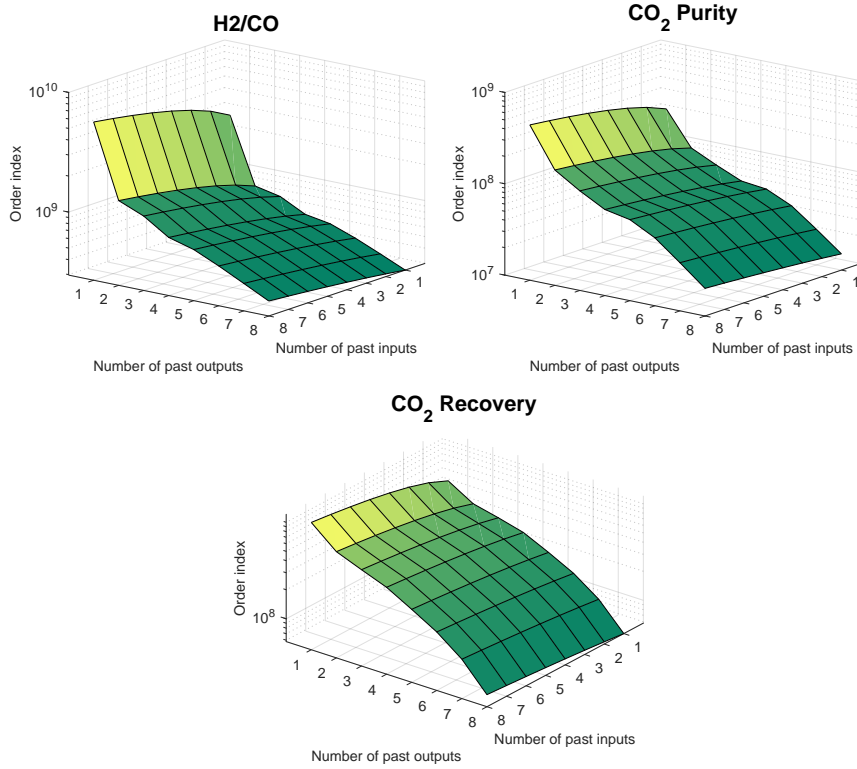


Figure 5.4: Lipschitz index values map.

Once the datasets are prepared and the predictor structure is defined, the next step is to build the DNN model using the training data. This process involves selecting an appropriate network architecture, configuring the layers, and tuning hyperparameters to achieve a good model performance. The model is trained using backpropagation and optimization algorithms that minimize the prediction error, ensuring a close fit to the system behavior and using the validation data to early stopping. The Hyperband optimization

method is employed to search for the selected set of hyperparameters as presented in Subsection 2.2.4. Table 5.5 presents the results obtained after the Hyperband optimization. This result is the backbone for this methodology’s next step, identifying the model’s uncertainties.

Table 5.5: Hyperband search space.

Hyperspace			
Hyperparameter	Output variables		
	$H_2/CO$	$CO_2$ purity	$CO_2$ recovery
Layer type	Dense	Dense	Dense
Number of dense layers	1 to 3	1 to 3	1 to 3
Number of output layers	1	1	1
Dense activation function	{‘tanh’, ‘relu’ and ‘linear’}	{‘tanh’, ‘relu’ and ‘linear’}	{‘tanh’, ‘relu’ and ‘linear’}
Neurons per layers	10 to 40, step 5	10 to 60, step 5	10 to 60, step 5
Dropout rate layers	0.25	0.25	0.25
Initial learning rate	$\{1 \times 10^{-4}, 1 \times 10^{-2}\}$ log sampling	$\{1 \times 10^{-4}, 1 \times 10^{-2}\}$ log sampling	$\{1 \times 10^{-4}, 1 \times 10^{-2}\}$ log sampling
Monitor metrics	MAE	MAE	MAE
Loss function	MSE	MSE	MSE
Objective function	min validation MAE	min validation MAE	min validation MAE
Max tested epoch	100	100	100
Batch size	256	256	256
Results			
Number of dense layers	3	3	3
Dense activation function	{tanh, tanh, tanh, linear}	{tanh, tanh, tanh, linear}	{tanh, tanh, tanh, linear}
Neurons by layers	{25, 25, 25, 26}	{35, 35, 35, 36}	{40, 40, 40, 41}
Params. per layer	{1275, 650, 650, 26}	{1785, 1260, 1260, 36}	{2040, 1640, 1640, 41}
Initial learning rate	0.002256	0.001560	0.001382
Number of parameters	2601	4341	5361

The final models’ architecture comprises sequential models, as shown in Table 5.5. With the DNN model established, MCMC training presented in Subsection 4.3.4 is applied to evaluate the parameter uncertainties. This technique offers a probabilistic framework to quantify the uncertainty associated with model parameters, accounting for model uncertainty and epistemic uncertainty. Also, the parameter uncertainty can be propagated to the model’s predictions with the same framework.

For the present case study, 20.000 posterior samples are sampled for each DNN model

parameter. In other words, the posterior DNN parameters PDF is a matrix {Number of parameters  $\times$  Number of samples}. Also, in this work, the Python library for MCMC estimation proposed by (MILES, 2019) was used to perform the MCMC.

Figures 5.5 to 5.7 present an analysis of two arbitrary parameters from each network, as well as a comprehensive overview of the chains. Consistent with optimization best practices, the Markov Chain Monte Carlo (MCMC) method used dimensionless network parameters. This strategy ensures that the values represented in the graphs are relative to the original training data.

Each figure encompasses subfigures (A) and (B), displaying the Markov chains for each parameter with 20,000 samples. This research assumes that the first 5,000 samples from the chain correspond to the burn-in phase, and as such, these samples are omitted from the propagation terms. Subfigures (C), (D), and (E) illustrate the marginalization of the PDFs for the selected parameters.

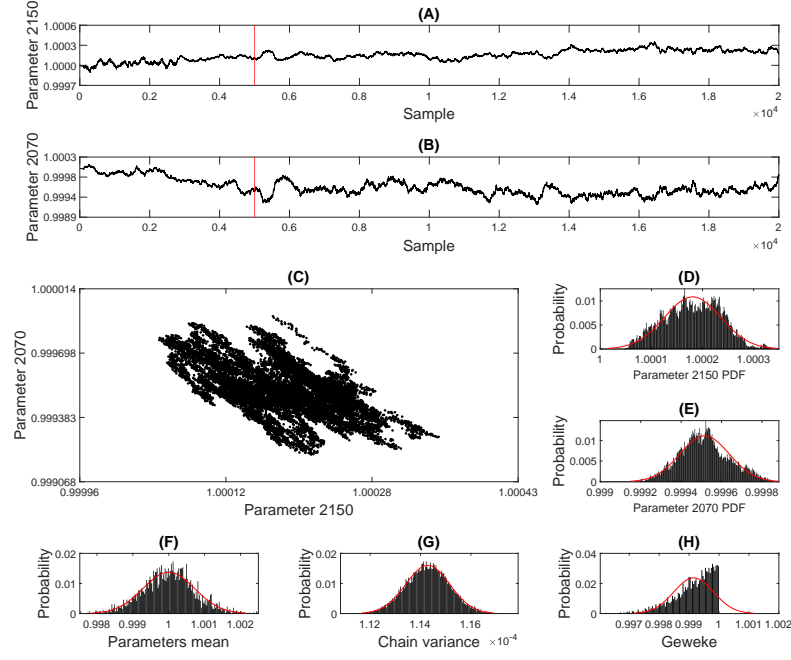
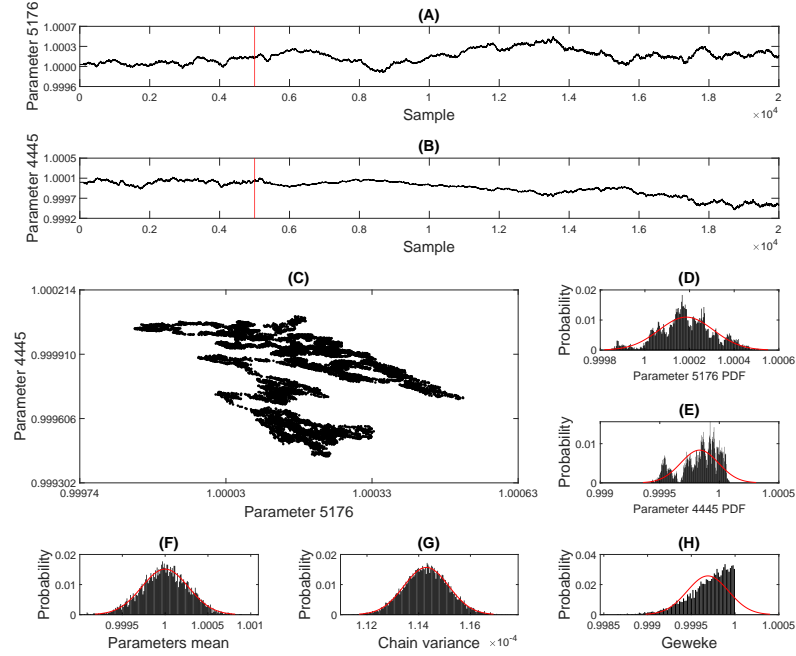
Non-informative prior distributions were chosen for the parameters in this study, a decision reinforced by the results shown in Figures 5.5 to 5.7. On the one hand, subfigures (D) and (E) of Figure 5.5 and (D) of Figure 5.7 demonstrate that the selected parameters are converging towards a normal distribution. Conversely, as depicted in Figure 5.6 (E) and Figure 5.7 (E), not all parameters follow this trend. This discrepancy indicates that using normal distributions for the priors might be premature. On another note, subfigure (F) represents the average of all dimensionless parameters within the network, while subfigure (G) reveals the variance of the chain throughout its construction. The variance is not centered around zero; however, its value is small, particularly considering it is a normalized parameter.

Finally, subfigure (H) in Figures 5.5 to 5.7 showcases the distribution of the Geweke parameter values, providing insight into the convergence of the chains. These parameters were calculated by comparing the means of the initial 10% of samples with the final 50% (BROOKS; ROBERTS, 1998). When a chain converges, its value tends towards 1. The subfigure (H) data suggests that the chains have satisfactorily converged. It is important to note that this evaluation was limited to two parameters to facilitate the analysis. However, the number of parameters in the models developed in this study (see. Table 5.5) precludes a thorough analysis of the convergence and marginalization of the PDFs. Nonetheless, confirming if the chains have converged is critical, and subfigure (H) can be utilized for this analysis.

### 5.2.3 Digital twining - Application tests

Following the initial steps, the system's digital twin (DT) is constructed and set to operate in parallel with the plant. The digital twin runs concurrently with the plant, receiving updates and adapting its performance through online learning. The second part of Figure 5.1 presents a flowchart illustrating the execution of the digital twin application. Hence, this section concerns the online deployment of the DT.

This Chapter proposes a feedback tracker (FBT) that compares the DT prediction confidence limits with the plant's measurements. The primary objective is to discern whether the violation of the limits stems from a persistent disturbance or natural dynamic

Figure 5.5:  $H_2/CO$  DNN parameters MCMC analysis.Figure 5.6:  $CO_2$  recovery DNN parameters MCMC analysis.

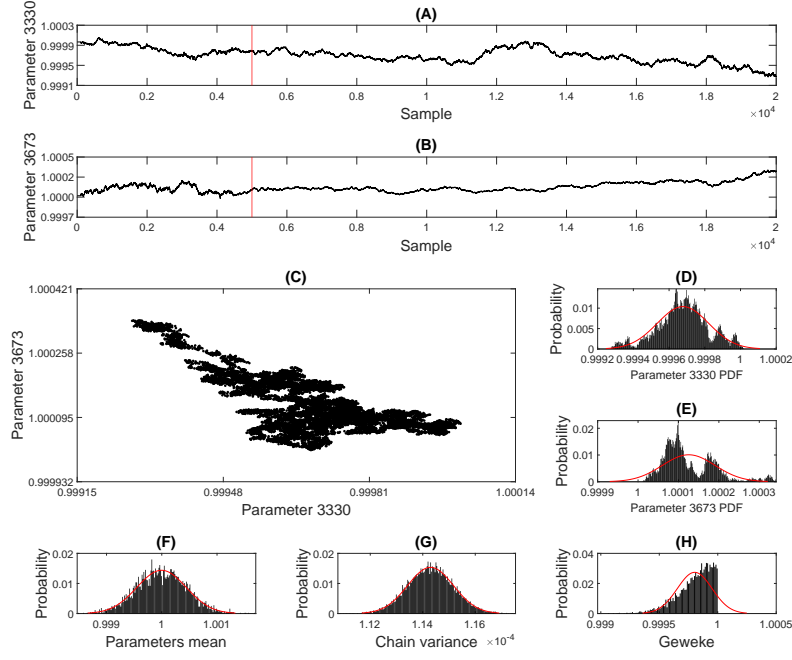


Figure 5.7:  $CO_2$  purity DNN parameters MCMC analysis.

variation within the DT and the system. This distinction is crucial for the appropriate employment of online learning, and it is even more critical in systems such as the PSA, where complex dynamic behavior is observed.

On the one hand, online learning can be a substantial computational burden, as it necessitates activating Bayesian inference to assess the new uncertainties each time the model is updated. This can lead to increased computational demands, which may not be feasible for real-time operation in specific scenarios.

On the other hand, online learning is essential for maintaining the accuracy and relevance of the digital twin, as it enables the model to adapt whenever it begins to drift away from the plant states. This dynamic adaptation is significant in complex, nonlinear systems, where the plant's behavior can change significantly over time.

Therefore, it is crucial to balance the computational burden associated with frequent online learning updates and the need to maintain an accurate representation of the plant states. This can be achieved by carefully defining the OL requirements and employing the FBT, allowing the digital twin to adapt to the plant's behavior while minimizing the computational effort required for updating the model and assessing the uncertainties.

$$FBT_i = \begin{cases} 1, & \text{if } y_i^m < y_{i,min}^{DT} \\ 1, & \text{if } y_i^m > y_{i,max}^{DT} \\ 0, & \text{otherwise} \end{cases} \quad (5.5)$$

$$FBT = \frac{\sum_{i=1}^N FBT_i}{N} \quad (5.6)$$

$$AC = \begin{cases} Active, & \text{if } FBT \geq R \\ Deactive, & \text{otherwise} \end{cases} \quad (5.7)$$

At the beginning of each loop, the system inputs are fed simultaneously into the plant and the digital twin. While the plant produces its actual value, the digital twin generates predictions and uncertainty assessment limits based on the same input signal. It is noteworthy that, over time, the usual degradation of the adsorbent in the PSA system is expected to cause alterations in the process behavior. Therefore, the prediction model for the system, represented here by the digital twin, will drift away from its original state.

The FTB then compares the plant's input and output signals with the DT limits during a specific time window ( $N$ ). Online learning remains inactive if the comparison meets the predetermined requirements ( $R$ ), and the loop continues. However, suppose the plant signal falls outside the DT limits for a specified number of samples, which could occur due to adsorbent degradation. In that case, OL is activated, and the DNN model undergoes retraining and reassessing the uncertainties. Hence, the weights matrixes are updated. During this time, the digital twin continues predicting the system, ensuring uninterrupted operation, and adjusting itself to the new system conditions brought about by the effects of adsorbent degradation. The degradation was added to the virtual plant within the model adsorption isotherm.

As new measurements are obtained, they are sent to a database that stores the plant's operational data. When the OL process is activated, the database is accessed to update the DNN model with the most recent information. This approach allows the digital twin to adapt to the plant's behavior efficiently while maintaining accurate representation and ensuring that the OL only occurs when necessary, thereby minimizing computational costs and maintaining prediction reliability.

Developing accurate DNN models for the PSA system can be challenging due to its complex dynamics. To overcome this, our work proposes a hybrid approach that combines actual and synthetic data to train and update the DT. Figure 5.1 incorporated past operating data and experimentally validated virtual plant data to update the DNN weights. Our approach ensures greater network convergence during the OL stage by reducing the uncertainty in system predictions.

To reduce the computational cost of the approach, the OL identifies a reduced number of parameter samples for retraining the DNN model. This number of samples is determined through sensitivity analysis, which helps balance the reduction of computational effort with the preservation of prediction reliability. As a result, the cycle resumes with an updated model that maintains accurate representation while minimizing the computational burden associated with continuous OL. This continuous process persists throughout the plant's operation, ensuring that the digital twin adapts to the plant's behavior while keeping the prediction reliability intact. Figures 5.8, 5.9, and 5.10 presents the final results obtained for a PSA campaign.

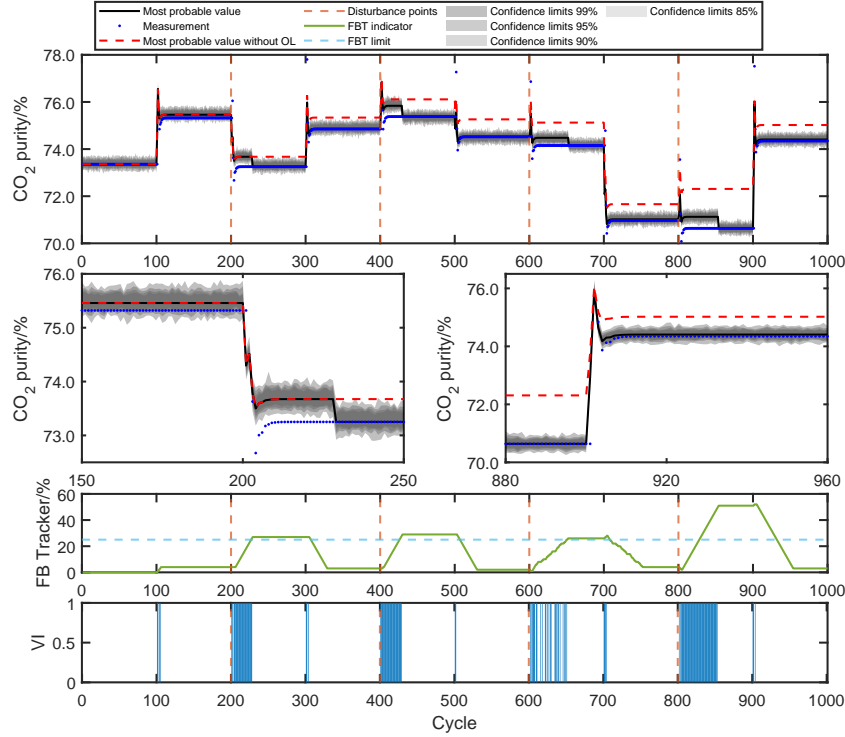


Figure 5.8: Digital Twin application to  $CO_2$  purity.

Figures 5.8, 5.9, and 5.10 illustrate the operation of the DT for three variables selected from the PSA system. In particular, Figures 5.8 and 5.9 present the results for the Purity and Recovery of  $CO_2$  captured by the system, while Figure 5.10 shows the behavior of the  $H_2/CO$  ratio, which is a crucial variable for the PSA Syngas process.

Figures 5.8, 5.9, and 5.10 show the prediction uncertainty region of the networks for different confidence limits. This analysis indicates that in addition to the most likely value, it is possible to judge whether the plant deviation is happening within the confidence region of the network prediction or outside it. In this sense, it is important to consider that TD revaluation actions should only be performed if the actual value is significantly out of line and outside the prediction confidence region, associated with exceeding the  $R$ , of Equation 5.7, established value of 20% of samples outside this region.

In these tests of the DT, four stages of catalyst degradation were considered for each of the three variables, corresponding to degradation levels of 5%, 10%, 15%, and 20%, respectively. These degradation points are indicated by vertical lines in Figures 5.8 to 5.10. The DT performance was evaluated by comparing the predicted and measured values, with and without the OL activated.

In Figures 5.8 and 5.9, the first graph shows that the DT accurately predicts the measured values in the initial 200 cycles. However, after the first stage of adsorbent degradation at cycle 200, the predicted values deviate from the measured values, indicating that the DT's confidence region has been exceeded. Nonetheless, the OL successfully restores the DT's prediction accuracy, while the non-OL scenario deviates from the actual



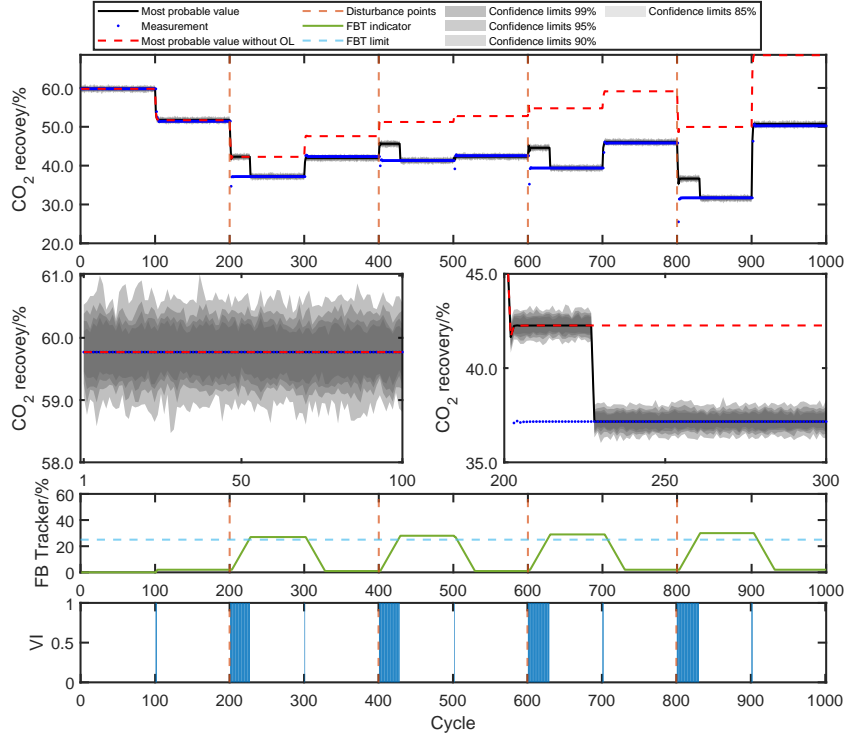


Figure 5.9: Digital Twin application to  $CO_2$  recovery.

measurements. This pattern repeats throughout the experiment.

Moreover, Figures 5.8 and 5.9 reveal the behavior of the FBT and the points at which the prediction limits were breached. Specifically, after the maximum value is exceeded, the OL is triggered, and the FBT indicator stops growing. This demonstrates the effectiveness of the OL in maintaining the system's performance within acceptable limits. After 100 samples, the FBT indicator decreases as the out-of-range samples are discarded.

As for Figure 5.10, the  $H_2CO$  ratio exhibits a weak dependence on the catalyst degradation and a strong correlation with the system's inputs. Therefore, the OL is activated after 900, given the more significant uncertainty in the network prediction for this variable than the variation caused by adsorbent degradation.

### 5.3 CONCLUSIONS

This chapter presented a comprehensive methodology for developing a digital twin of a PSA system using DNNs, Bayesian inference, and online learning. By combining these advanced techniques, the digital twin can accurately represent the PSA system's complex and dynamic behavior while continuously adapting to changes through online learning.

The proposed methodology addresses several challenges associated with modeling PSA systems, such as AI-based modeling of periodic dynamics for prediction purposes, model uncertainty quantification, and reliable online learning. By incorporating Bayesian infer-

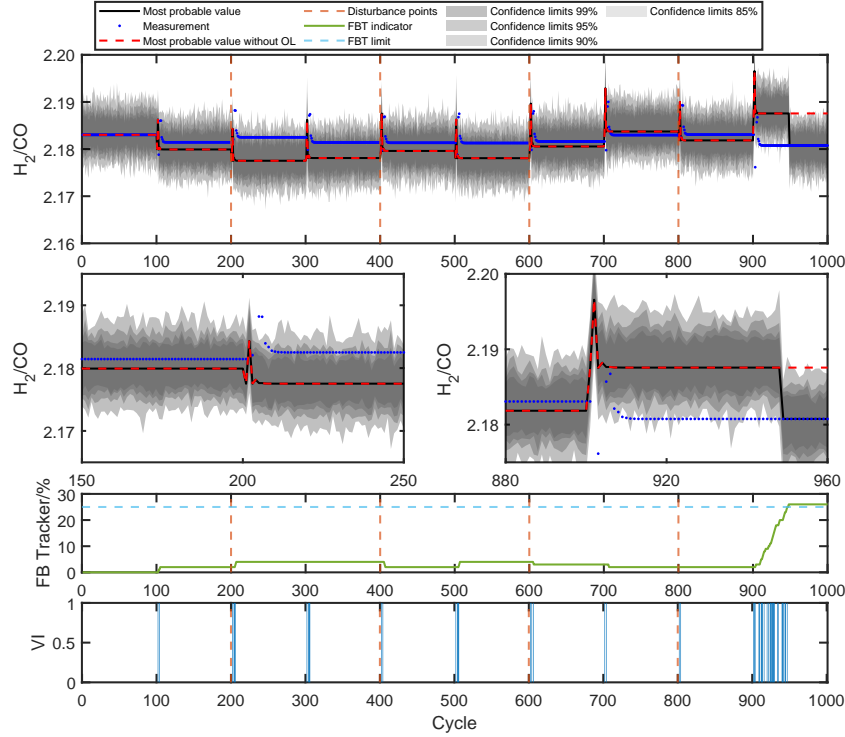


Figure 5.10: Digital Twin application to  $H_2$  and  $CO$  ratio.

ence and Markov Chain Monte Carlo methods, the digital twin can estimate and account for the uncertainties in data and model parameters, leading to more reliable and robust predictions.

The proposed digital twin allows it to serve as an environment for control applications, optimization, and monitoring of the system operation in real-time. This approach allows performing performance tests on the digital twin before actual implementation in the real system. This allows new operating points to be acquired and sent to the real system with minimal disruption to production.

Moreover, online learning enables the digital twin to adapt and improve its performance over time, providing enhanced decision-making capabilities for process monitoring, optimization, and control. Integrating a novel feedback tracker further strengthens the digital twin's representativity of the PSA system, ensuring the model remains responsive and accurate throughout its operation.

The presented methodology offers a promising foundation for developing AI-based digital twins in various industrial applications, particularly within Industry 4.0. By addressing the challenges of modeling complex and dynamic systems and quantifying and managing uncertainty, the proposed digital twin can enhance the safety, reliability, and efficiency of pressure swing adsorption processes and other complex industrial systems.

### 6.1 FINAL REMARKS

This thesis presented methods for constructing and validating phenomenological and model-based dynamic models and Digital twins using Bayesian inference to assess uncertainty. The results for each chapter can be separated as:

Chapter 2 - It presented the general methodology proposed by the article and tested two of the four main hypotheses evaluated in the thesis. The results indicate that the hypothesis that uncertainty can be applied to both phenomenological and data-driven models. Furthermore, he showed that the development of artificial intelligence models can be significantly facilitated using synthetic data. Also, Chapter 2 presented a case study focusing on two soft sensors designed to monitor the temperature and viscosity of a polymerization reactor. The simulation results demonstrated that the predictions from the soft sensors are statistically equal to the validation data in both dynamic and stationary regimes. The results indicate the proposed methodology's consistency and reliability for evaluating uncertainty in scientific machine learning models.

Chapter 3 showed that the initial hypothesis to be tested is valid for phenomenological models when these models are constructed and the parameters are estimated from experimental data. The proposed method, using Bayesian inference, proved to be effective in obtaining the posterior marginal distribution of the parameters of the nonlinear model. Additionally, the propagation of uncertainty to the model prediction allowed a deeper understanding of the model's prediction capacity. The Bayesian approach was applied to validate an ESP system phenomenological model using data from installed sensors to update the equipment's operating parameters. The validation concluded that the model could represent the experimental data within the existing uncertainties, indicating the method's efficacy in building the joint PDF of model parameters and obtaining confidence bounds. As a result, the validated model is a suitable candidate for serving as a surrogate for the experimental apparatus.

Chapter 4 - The proposed methods of Chapters 2, 3 are combined to provide the uncertainty assessment of DNN models to the ESP system. This chapter tests the first two hypotheses again to obtain data-driven models. Thus, the synthetic data generated by the validated nonlinear simulator and Bayesian inference is applied to assess the uncertainty of DNN models for ESP systems, evaluating their performance using experimental data in real-case scenarios. The study found that a NARX model using a DNN with fully connected layers accurately represents the behavior of the ESP system. The uncertainty assessment confirmed the alignment between the DNN models, the nonlinear model, and the experimental data, demonstrating the successful prediction of crucial ESP system parameters.

Chapter 5 – Developed a digital twin for a PSA system, integrating DNNs, Bayesian inference, and online learning to represent complex behavior and facilitate real-time applications. In this chapter, all hypotheses are tested together. Through analysis of the results, it is possible to observe that assessing the uncertainty of prediction models in digital twins provides valuable information regarding the need to update and maintain the models. The methodology used in Chapter 5 is based on the previous chapter’s proposed methodologies. The digital twin developed proved capable of addressing challenges associated with modeling PSA systems and enhancing decision-making capabilities for real-time control, optimization, and monitoring. This approach is a promising foundation for developing AI-based digital twins in various industrial applications, particularly within Industry 4.0, contributing to enhanced safety, reliability, and efficiency.

Some of the contributions are listed as follows:

- Application of Markov Chain Monte Carlo, synthetic data generation, neural network structure identification, Monte Carlo simulation training, and methodology validation for evaluating uncertainty in Scientific Machine Learning Models. These topics were connected in a comprehensive five-step approach for addressing uncertainties from model structure and data.
- Implement a Bayesian approach for validating ESP system models using sensor data. Also, it introduced an innovative approach to uncertainty assessment of ESP systems modeling. This methodology successfully validates NARX DNN models predicting crucial ESP system parameters. Systematic treatment of uncertainty and integrated parameter acquisition coupled with a practical construction of the joint PDF of model parameters and confidence bounds. This methodology was tested in a case study building Soft Sensors for monitoring a polymerization reactor.
- Integration of DNNs, Bayesian inference, and online learning for dynamic behavior representation and adaptability to provide a comprehensive methodology for developing a digital twin of a PSA system.
- Provide details about further robust SciML applications in the ESP and PSA field, also contributing to the foundation for Industry 4.0 applications, enhancing safety, reliability, and efficiency.

## 6.2 FUTURE WORK

Based on the results and methodologies presented in this thesis, the following future works are suggested for new research and perspectives:

- Investigate the application and adaptability of the proposed methodology to other scientific domains.
- Explore enhancements in uncertainty quantification and the incorporation of additional data sources.
- Constructing and validating multiphase and non-isothermal models.

- Investigate the method's applicability to different ESP systems and operating conditions.
- Explore the development of digital twins, soft sensors, control, optimization, and production monitoring applications.
- Evaluate the potential enhancements and adaptations of the NARX DNN models for varied scenarios and parameters.
- Expand the application of the digital twin methodology to various industrial applications and complex systems.
- Explore advancements in modeling techniques, online learning algorithms, and uncertainty management to enhance digital twins further.
- Investigate the impact of real-time adaptability on system safety, reliability, and efficiency in diverse industrial settings.
- Develop and investigate the application of the methodology for hybrid models.



## BIBLIOGRAPHY

ABDAR, M. et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, v. 76, p. 243–297, 12 2021. ISSN 15662535. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1566253521001081>.

ABDOLRASOL, M. G. M. et al. Artificial neural networks based optimization techniques: A review. *Electronics*, v. 10, n. 21, 2021. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/10/21/2689>.

ABDURAKIPOV, S. Increasing the efficiency of electric submersible pumps by using big data processing and machine learning technologies. *Journal of Physics: Conference Series*, v. 2119, p. 012109, 12 2021. ISSN 1742-6588. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/2119/1/012109>.

ALAMU, O. A. et al. Esp data analytics: Use of deep autoencoders for intelligent surveillance of electric submersible pumps. In: . Offshore Technology Conference, 2020. v. 2020-May. ISBN 9781613997079. ISSN 01603663. Cited By 0; Conference of Offshore Technology Conference 2020, OTC 2020 ; Conference Date: 4 May 2020 Through 7 May 2020; Conference Code:160167. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85086268361&partnerID=40&md5=08dfa66d2eab6415d6a43f9586823ecf>.

ALVAREZ, L. A.; ODLOAK, D. Optimization and control of a continuous polymerization reactor. *Brazilian Journal of Chemical Engineering*, v. 29, p. 807–820, 2012. ISSN 0104-6632. Disponível em: [www.abeq.org.br/bjche](http://www.abeq.org.br/bjche).

BARBOSA, T. D. S. *Ambiente para avaliação de controladores Fuzzy aplicados ao método de elevação artificial por bombeio centrífugo submerso*. 116 p. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2011.

BARD, Y. *Nonlinear Parameter Estimation*. [S.l.]: Academic Press, 1974. 341 p. ISBN 9780120782505.

BATISTA, E. S. *Desenvolvimento de uma Ferramenta Computacional para Aplicação no Método de Elevação por Bombeio Centrífugo Submerso*. 81 p. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2009.

BINDER, B.; JOHANSEN, T.; IMSLAND, L. Improved predictions from measured disturbances in linear model predictive control. *Journal of Process Control*, Elsevier Ltd, v. 75, p. 86–106, 2019. ISSN 09591524.

BINDER, B. J. T.; KUFOALOR, D. K. M.; PAVLOV, J. Embedded Model Predictive Control for an Electric Submersible Pump on a Programmable Logic Controller. *IEEE Conference on Control Applications (CCA)*, n. 2, p. 579 – 585, 2014.

BINDER, B. J. T.; PAVLOV, A.; JOHANSEN, T. A. Estimation of flow rate and viscosity in a well with an electric submersible pump using moving horizon estimation. *IFAC-PapersOnLine*, Elsevier Ltd., v. 28, n. 6, p. 140–146, 2015.

BIPM et al. *Evaluation of measurement data - Guide to the expression of uncertainty in measurement. Joint committee for guides in metrology*. [S.l.], 2008. 134 p. (JCGM 100:2008).

BIPM et al. *Evaluation of measurement data — Supplement 1 to the “Guide to the expression of uncertainty in measurement” — Propagation of distributions using a Monte Carlo method*. Joint Committee for Guides in Metrology - JCGM 101:2008, 2008. 90 p. Disponível em: <http://www.bipm.org/utis/common/documents/jcgm/JCGM\101\2008\E.pdf>.

BIPM et al. Evaluation of measurement data — supplement 1 to the “guide to the expression of uncertainty in measurement” — propagation of distributions using a monte carlo method. *Evaluation*, Joint Committee for Guides in Metrology - JCGM 101:2008, JCGM 101:2, p. 90, 2008. Disponível em: <http://www.bipm.org/utis/common/documents/jcgm/JCGM\101\2008\E.pdf>.

BIPM et al. Evaluation of measurement data – supplement 2 to the “guide to the expression of uncertainty in measurement” – models with any number of output quantities. Joint Committee for Guides in Metrology - JCGM 102:2011, p. 80, 2011.

BIPM et al. *International vocabulary of metrology - Basic and general concepts and associated terms (VIM)*. Joint committee for guides in metrology. [S.l.], 2012. 108 p.

BISCHL, B. et al. *Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges*. 2021.

BROOKS, S.; ROBERTS, G. Assessing convergence of markov chain monte carlo algorithms. v. 8, 08 1997.

BROOKS, S. P.; ROBERTS, G. O. Assessing convergence of markov chain monte carlo algorithms. *Statistics and Computing*, v. 8, p. 319–335, 1998. Disponível em: <https://doi.org/10.1023/A:1008820505350>.

CHOLLET, F. et al. *Keras*. 2015. <https://keras.io>.

CHOUBEY, S.; KARMAKAR, G. P. Artificial intelligence techniques and their application in oil and gas industry. *Artificial Intelligence Review*, Springer Netherlands, v. 54, p. 3665–3683, 6 2021. ISSN 0269-2821. Disponível em: <https://doi.org/10.1007/s10462-020-09935-1https://link.springer.com/10.1007/s10462-020-09935-1>.



CHUANG, K. V.; KEISER, M. J. Adversarial controls for scientific machine learning. *ACS Chemical Biology*, v. 13, p. 2819–2821, 10 2018. ISSN 1554-8929.

COSTA, E. et al. A bayesian approach to the dynamic modeling of esp-lifted oil well systems: An experimental validation on an esp prototype. *Journal of Petroleum Science and Engineering*, Elsevier B.V., v. 205, p. 108880, 10 2021. ISSN 09204105. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85105312861&doi=10.1016/j.petrol.2021.108880&partnerID=40&md5=fedd41e9fb43e09ffdda8ab1246d03d0https://linkinghub.elsevier.com/retrieve/pii/S0920410521005416>.

COSTA, E. A. et al. A robust learning methodology for uncertainty-aware scientific machine learning models. *Mathematics*, v. 11, p. 74, 12 2022. ISSN 2227-7390.

COSTA, R. d. O. *CONTROLE APLICADO A POÇOS COM MÉTODO DE ELEVAÇÃO BOMBEIO CENTRÍFUGO SUBMERSO*. 138 p. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2012.

COSTA, R. O. et al. Override fuzzy controller applied at the electrical submersible pumping method. In: *Rio Pipeline Conference and Exposition, Technical Papers*. [S.l.]: Instituto Brasileiro de Petróleo, 2013. v. 2013-Sept, p. 10. ISSN 24472069.

DAS, W.; KHANNA, S. A robust machine learning based framework for the automated detection of adhd using pupillometric biomarkers and time series analysis. *Scientific Reports*, v. 11, p. 16370, 12 2021. ISSN 2045-2322.

DELOU, P. d. A. et al. Model Predictive Control with Adaptive Strategy Applied to an Electric Submersible Pump in a Subsea Environment. *IFAC-PapersOnLine*, Elsevier B.V., v. 52, n. 1, p. 784–789, 2019. ISSN 24058963.

FILHO, F. G. d. A.; MAITELLI, A. L. A Neuro-Fuzzy Control Scheme for the Electrical Submersible Pumping in Oilfield Systems. *IEEE Latin America Transactions*, v. 15, n. 11, p. 2077–2083, 11 2017.

GAIKWAD, A. et al. Heterogeneous sensing and scientific machine learning for quality assurance in laser powder bed fusion – a single-track study. *Additive Manufacturing*, v. 36, p. 101659, 12 2020. ISSN 22148604.

GAL, Y.; GHAHRAMANI, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. 6 2015. Disponível em: <http://arxiv.org/abs/1506.02142>.

GAMERMAN, D.; LOPES, H. F. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. 2. ed. [S.l.]: Chapman and Hall/CRC, 2006. 343 p. ISBN 1-58488-587-4.

GELMAN, A. *Bayesian data analysis 3rd ed.* [S.l.: s.n.], 2015. 33-36 p. ISSN 16130073. ISBN 9788578110796.

GELMAN, A. et al. Bayesian data analysis third edition (with errors fixed as of 13 february 2020). p. 677, 2013. Disponível em: [http://files/25/Gelmanm.fl.-BayesianDataAnalysisThirdedition\(witherrors.pdf\)](http://files/25/Gelmanm.fl.-BayesianDataAnalysisThirdedition(witherrors.pdf)).

GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. [S.l.]: O'Reilly Media, Inc., 2019. 801 p. ISBN 9781492032649.

GEWEKE, J. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In: \_\_\_\_\_. *Bayesian Statistics 4*. Oxford University PressOxford, 1992. p. 169–194. ISBN 9781383023756. Disponível em: <http://dx.doi.org/10.1093/oso/9780198522669.003.0010>.

HAARIO, H. et al. DRAM: Efficient adaptive MCMC. *Statistics and Computing*, v. 16, n. 4, p. 339–354, 2006. ISSN 09603174.

HAARIO, H. et al. Dram: Efficient adaptive mcmc. *Statistics and Computing*, v. 16, p. 339–354, 2006. ISSN 09603174.

HAARIO, H.; SAKSMAN, E.; TAMMINEN, J. An adaptive metropolis algorithm. *Bernoulli*, v. 7, p. 223–242, 2001. ISSN 13507265.

HAARIO, H.; SAKSMAN, E.; TAMMINEN, J. An adaptive Metropolis algorithm. *Bernoulli*, v. 7, n. 2, p. 223–242, 2001. ISSN 13507265.

HAYKIN, S. *Neural networks and learning machines*. 2. ed. [S.l.]: Pearson Prentice Hall, 1999. 938 p. ISBN 9780131471399.

HE, X.; ASADA, H. New method for identifying orders of input-output models for non-linear dynamic systems. In: . [S.l.]: Publ by IEEE, 1993. p. 2520–2523. ISBN 0780308611.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997.

HODSON, T. O. Root-mean-square error (rmse) or mean absolute error (mae): when to use them or not. *Geoscientific Model Development*, v. 15, n. 14, p. 5481–5487, 2022. Disponível em: <https://gmd.copernicus.org/articles/15/5481/2022/>.

KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. arXiv, 2014. Disponível em: <https://arxiv.org/abs/1412.6980>.

KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. 2017.

KLEIJNEN, J. P. C. *Verification and validation of simulation models*. 1995. 145-162 p.

KRISHNAMOORTHY, D. et al. Modelling and Robustness Analysis of Model Predictive Control for Electrical Submersible Pump Lifted Heavy Oil Wells. *IFAC-PapersOnLine*, Elsevier B.V., v. 49, n. 7, p. 544–549, 2016.

KULLICK, J.; HACKL, C. M. Dynamic modeling and simulation of deep geothermal electric submersible pumping systems. *Energies*, v. 10, n. 10, p. 1–37, 10 2017. ISSN 19961073.

LASTRA, R. Electrical Submersible Pump Digital Twin, the Missing Link for Successful Condition Monitoring and Failure Prediction. In: *Abu Dhabi International Petroleum Exhibition & Conference*. [S.l.]: Society of Petroleum Engineers, 2019. p. 1–10. ISBN 9781613996720.

LI et al. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, v. 18, p. 1–52, 3 2016. ISSN 15337928. Disponível em: <http://arxiv.org/abs/1603.06560>.

LI, H. et al. Applications of artificial intelligence in oil and gas development. *Archives of Computational Methods in Engineering*, Springer Netherlands, v. 28, p. 937–949, 5 2021. ISSN 1134-3060. Disponível em: <https://doi.org/10.1007/s11831-020-09402-8><https://link.springer.com/10.1007/s11831-020-09402-8>.

LI, J. Z. *Principled Approaches to Robust Machine Learning and Beyond*. P.h.D — Department of Electrical Engineering and Computer Science, 9 2018.

LI, L. et al. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, v. 18, p. 1–52, 3 2016. ISSN 15337928. Disponível em: <http://arxiv.org/abs/1603.06560>.

MARTINS, M. A. et al. Artificial intelligence-oriented economic non-linear model predictive control applied to a pressure swing adsorption unit: Syngas purification as a case study. *Separation and Purification Technology*, v. 276, p. 119333, 12 2021. ISSN 13835866.

MENEZES, F. C. et al. Application of uncertainty analysis of artificial neural networks for predicting coagulant and alkalizer dosages in a water treatment process. *Brazilian Journal of Chemical Engineering*, v. 35, p. 1369–1381, 12 2018. ISSN 1678-4383. Disponível em: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0104-66322018000401369&tlng=en](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-66322018000401369&tlng=en).

MIGON, S. H.; GAMERMAN, D.; LOUZADA, F. *Statistical Inference: An Integrated Approach*. 2. ed. [S.l.]: CRC Press, 2014. 385 p.

MILES, P. pymcmcstat: A python package for bayesian inference using delayed rejection adaptive metropolis. *Journal of Open Source Software*, v. 4, p. 1417, 6 2019. ISSN 2475-9066.

MOHAMMADZAHERI, M. et al. Modelling of petroleum multiphase flow in electrical submersible pumps with shallow artificial neural networks. *Ships and Offshore Structures*, Taylor and Francis Ltd., v. 15, p. 174–183, 2020. ISSN 17445302. Cited By 7. Disponível em: [10.1080/17445302.2019.1605959](https://doi.org/10.1080/17445302.2019.1605959).

MOOS, J. et al. Robust reinforcement learning: A review of foundations and recent advances. *Machine Learning and Knowledge Extraction*, MDPI AG, v. 4, n. 1, p. 276–315, mar. 2022. ISSN 2504-4990. Disponível em: <http://dx.doi.org/10.3390/make4010013>.

NELLES, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes*. Springer International Publishing, 2020. ISBN 9783030474393. Disponível em: <http://dx.doi.org/10.1007/978-3-030-47439-3>.

NOGUEIRA, I. B. et al. A novel nested loop optimization problem based on deep neural networks and feasible operation regions definition for simultaneous material screening and process optimization. *Chemical Engineering Research and Design*, v. 180, p. 243–253, 4 2022. ISSN 02638762.

NOGUEIRA, I. B. et al. Dynamics of a True Moving Bed Reactor: Synthesis of n-Propyl Propionate and an alternative optimization method. *Chemical Engineering and Processing - Process Intensification*, v. 148, p. 107821, feb 2020. ISSN 02552701. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S025527011931400X>.

NOGUEIRA, I. B. R. et al. Big data-based optimization of a pressure swing adsorption unit for syngas purification: On mapping uncertainties from a metaheuristic technique. *Industrial & Engineering Chemistry Research*, v. 59, p. 14037–14047, 8 2020. ISSN 0888-5885.

OLIVEIRA, L. M. C. et al. Artificial intelligence and cyber-physical systems: A review and perspectives for the future in the chemical industry. *AI*, v. 2, p. 429–443, 9 2021. ISSN 2673-2688.

O'MALLEY, T. et al. *Keras Tuner*. 2019. <https://github.com/keras-team/keras-tuner>.

OZDEMIR, A. A.; GUMUSSOY, S. Transfer function estimation in system identification toolbox via vector fitting. *IFAC-PapersOnLine*, Elsevier B.V., v. 50, p. 6232–6237, 2017. ISSN 24058963. Disponível em: <https://doi.org/10.1016/j.ifacol.2017.08.1026>.

PAVLOV, A. et al. Modelling and model predictive control of oil wells with Electric Submersible Pumps. In: *2014 IEEE Conference on Control Applications (CCA)*. [S.l.]: IEEE, 2014. p. 586–592.

POSSOLO, A. Copulas for uncertainty analysis. *Metrologia*, v. 47, n. 3, p. 262–271, 2010. ISSN 00261394.

PSAROS, A. F. et al. *Uncertainty Quantification in Scientific Machine Learning: Methods, Metrics, and Comparisons*. arXiv, 2022. Disponível em: <https://arxiv.org/abs/2201.07766>.

RACKAUCKAS, C. et al. *Universal Differential Equations for Scientific Machine Learning*. arXiv, 2020. Disponível em: <https://arxiv.org/abs/2001.04385>.

REBELLO, C. M. et al. Machine learning-based dynamic modeling for process engineering applications: A guideline for simulation and prediction from perceptron to deep learning. *Processes*, v. 10, p. 250, 1 2022. ISSN 2227-9717.

REBELLO, C. M. et al. A novel standpoint of pressure swing adsorption processes multi-objective optimization: An approach based on feasible operation region mapping. *Chemical Engineering Research and Design*, v. 178, p. 590–601, 2 2022. ISSN 02638762.

REGUFE, M. J. et al. Syngas purification by porous amino- functionalized titanium terephthalate mil-125. *Energy & Fuel*, v. 29 (7), p. 4654–4664, 2015.

RIBEIRO, A. M.; SANTOS, J. C.; RODRIGUES, A. E. Pressure swing adsorption for CO<sub>2</sub> capture in Fischer-Tropsch fuels production from biomass. *Adsorption*, v. 17, n. 3, p. 443–452, jun 2011. ISSN 0929-5607. Disponível em: <http://link.springer.com/10.1007/s10450-010-9280-8>.

RØNNING, R. *Automatic Start-up and Control of Artificially Lifted Wells*. 122 p. Master — Norwegian University of Science and Technology, 2011. Disponível em: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/261113>.

ROY, V. Convergence diagnostics for markov chain monte carlo. *Annual Review of Statistics and Its Application*, Annual Reviews, v. 7, n. 1, p. 387–412, mar. 2020. ISSN 2326-831X. Disponível em: <http://dx.doi.org/10.1146/annurev-statistics-031219-041300>.

SCHWAAB, M.; PINTO, J. C. *Análise de dados experimentais, I: Fundamentos de Estatística e Estimação de parâmetros*. [S.l.]: E-Pappers, 2007. 462 p.

SHARMA, R.; GLEMMESTAD, B. Optimal control strategies with nonlinear optimization for an Electric Submersible Pump lifted oil field. *Modeling, Identification and Control: A Norwegian Research Bulletin*, v. 34, n. 2, p. 55–67, 2013.

SHARMA, R.; GLEMMESTAD, B. Uncertainty and Sensitivity Analysis for a Model of an Electric Submersible Pump Lifted Oil Field. *International Journal of Modeling and Optimization*, v. 4, n. 4, p. 278–286, 8 2014.

SILVA, M. P. et al. Water vapor harvesting by a (P)TSA process with MIL – 125(Ti)<sub>N</sub>H<sub>2</sub> as adsorbent. *Separation and Purification Technology*, v. 237, p. 116336, apr 2020. ISSN 13835866. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1383586619316119>.

STRAWDERMAN, R. L.; GAMERMAN, D. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. 2. ed. [S.l.]: Chapman and Hall/CRC, 2006. 343 p. ISBN 1584885874.

SUBRAVETI, S. G. et al. Physics-based neural networks for simulation and synthesis of cyclic adsorption processes. *Industrial & Engineering Chemistry Research*, v. 61, p. 4095–4113, 3 2022. ISSN 0888-5885.

TAKACS, G. *Electrical Submersible Pumps Manual*. Elsevier, 2018. 578 p. ISBN 9780128145708. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/C20170013083>.

TURZO, Z.; TAKACS, G.; ZSUGA, J. Equations correct centrifugal pump curves for viscosity. *Oil and Gas Journal*, v. 98, p. 57–67, 05 2000.

ZHU, J.; ZHANG, H.-q. A Review of Experiments and Modeling of Gas-Liquid Flow in Electrical Submersible Pumps. *Energies*, v. 11, n. 1, p. 180, jan 2018. ISSN 1996-1073. Disponível em: <http://www.mdpi.com/1996-1073/11/1/180>.

## APPENDIX A

### SUPPLEMENTARY ANALYSIS OF MCMC CHAINS

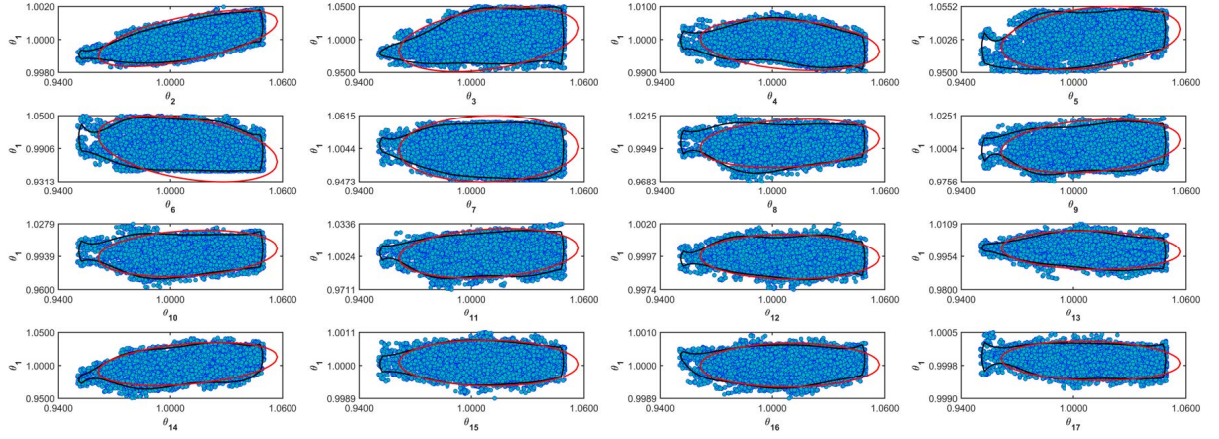


Figure A.1: Coverage regions of parameters 01.  
(—) Gaussian region; (—) Possolo (2010) region.

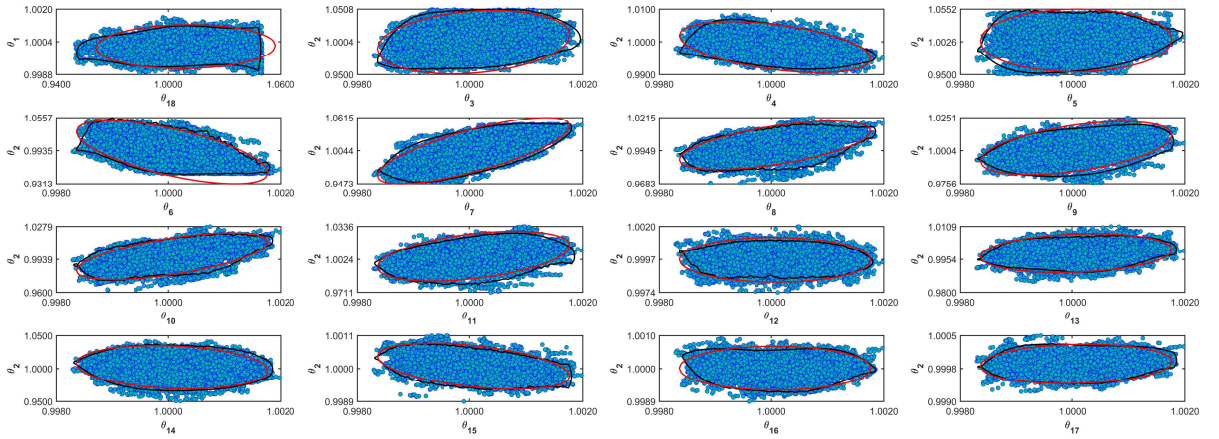


Figure A.2: Coverage regions of parameters 02.  
(—) Gaussian region; (—) Possolo (2010) region.



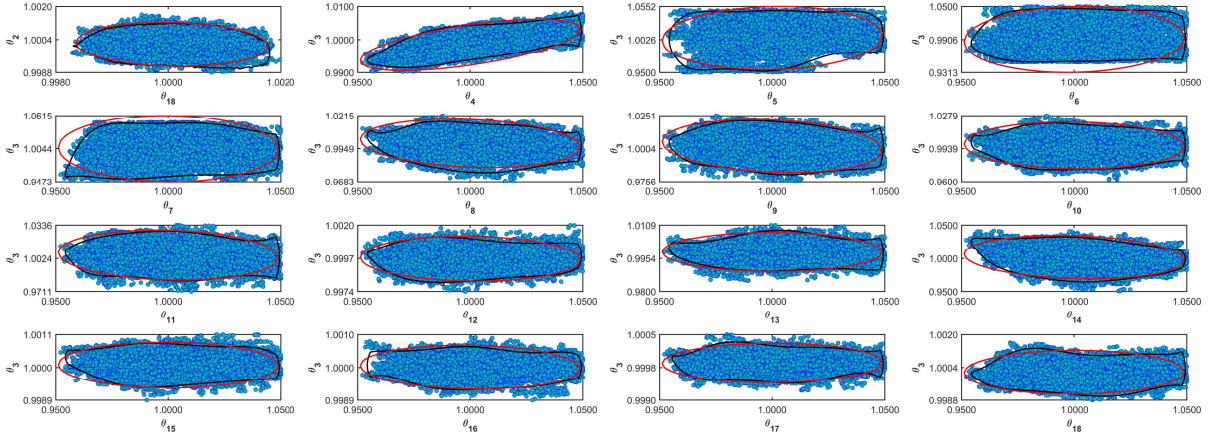


Figure A.3: Coverage regions of parameters 03.  
 (—) Gaussian region; (—) Possolo (2010) region.

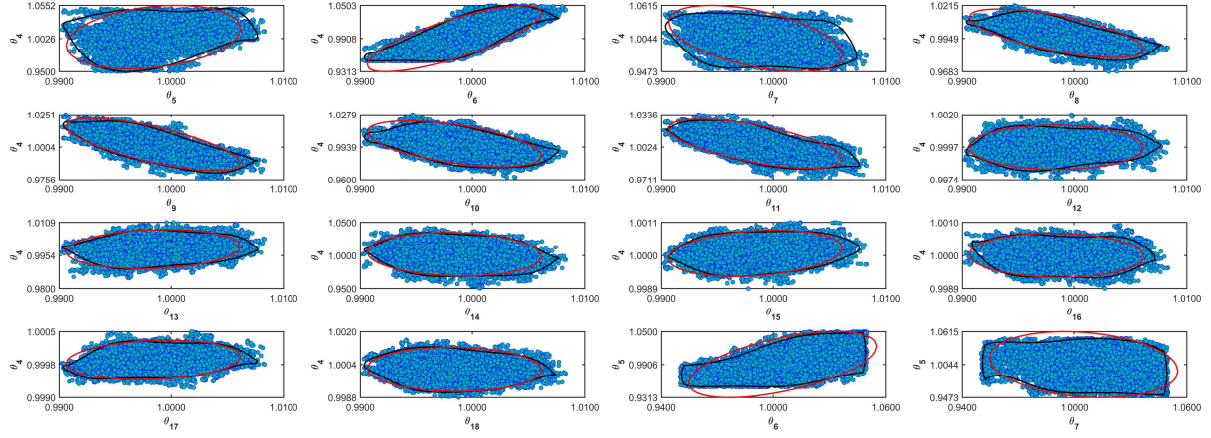


Figure A.4: Coverage regions of parameters 04.  
 (—) Gaussian region; (—) Possolo (2010) region.

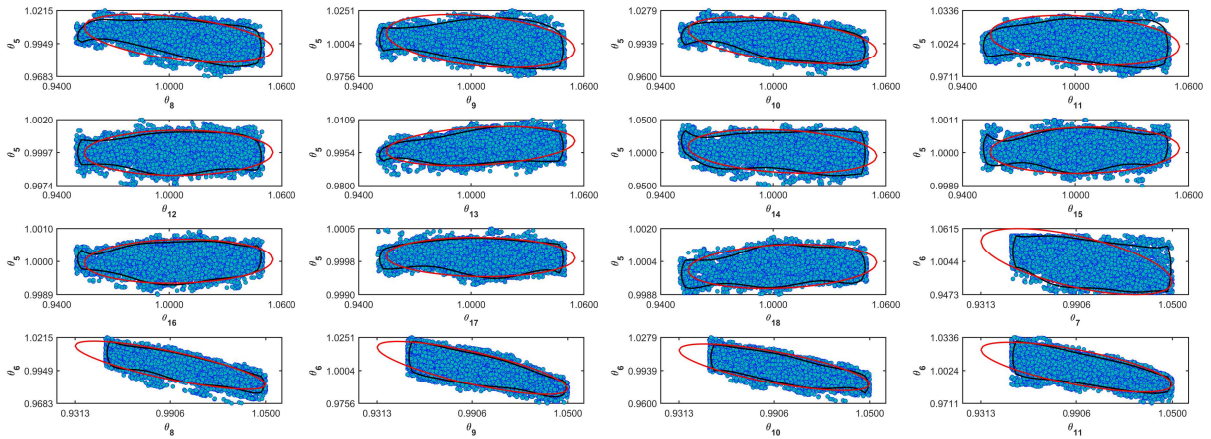


Figure A.5: Coverage regions of parameters 05.  
 (—) Gaussian region; (—) Possolo (2010) region.



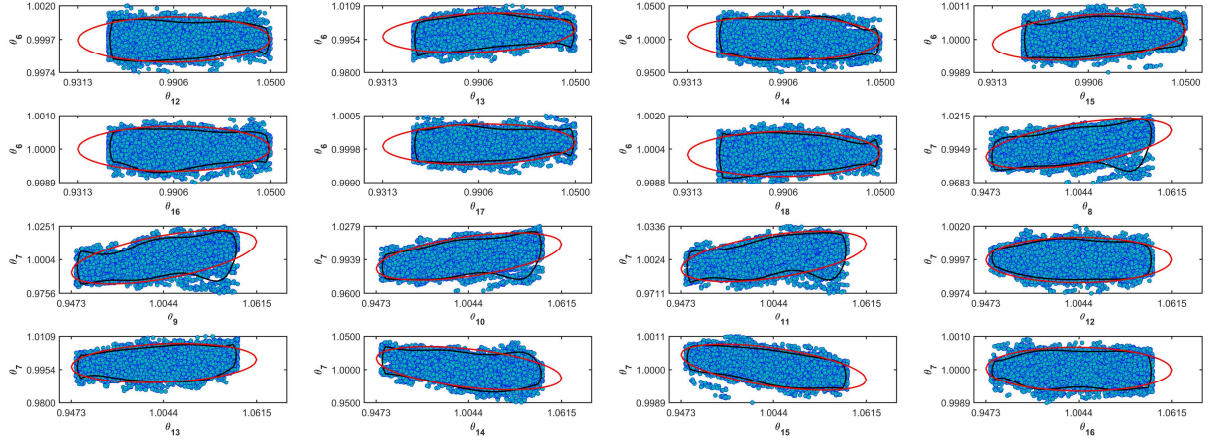


Figure A.6: Coverage regions of parameters 06.  
 (—) Gaussian region; (—) Possolo (2010) region.

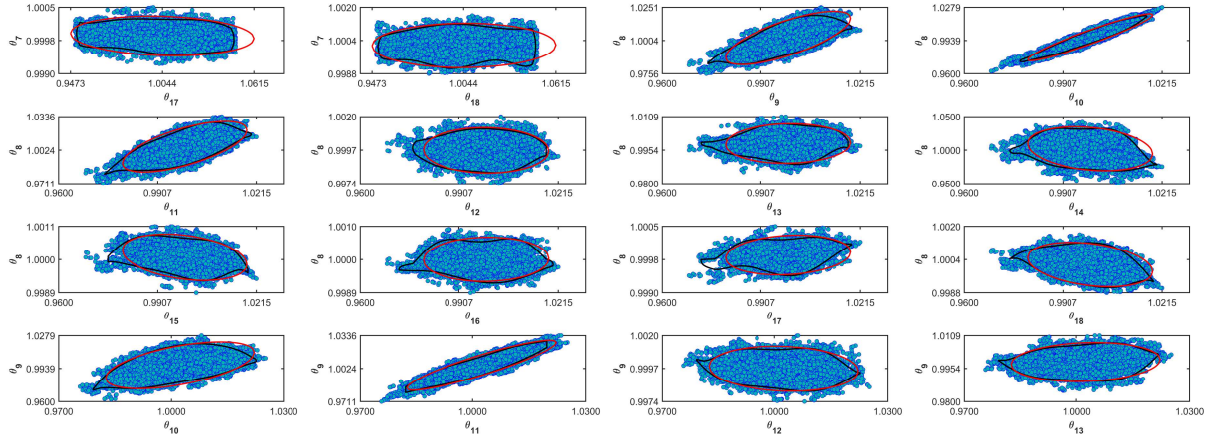


Figure A.7: Coverage regions of parameters 07.  
 (—) Gaussian region; (—) Possolo (2010) region.

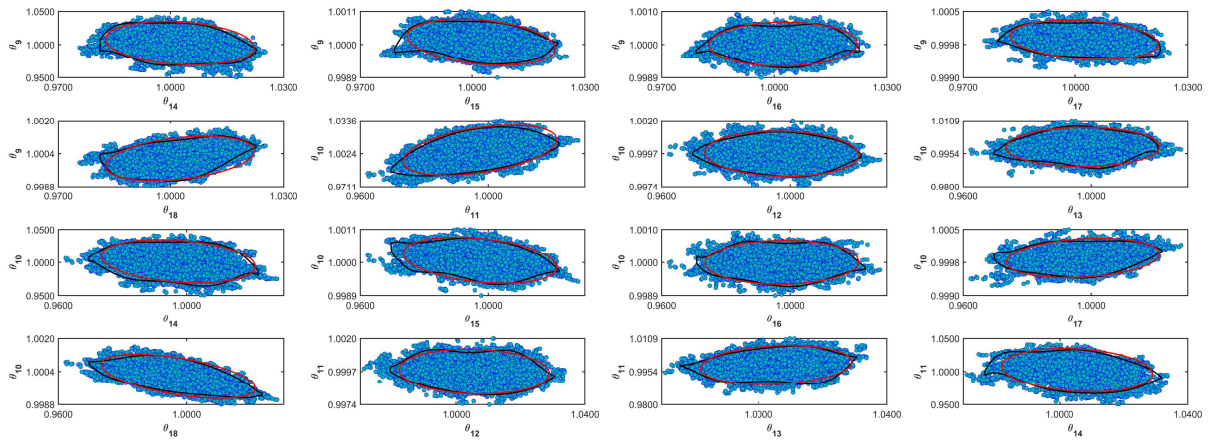


Figure A.8: Coverage regions of parameters 08.  
 (—) Gaussian region; (—) Possolo (2010) region.

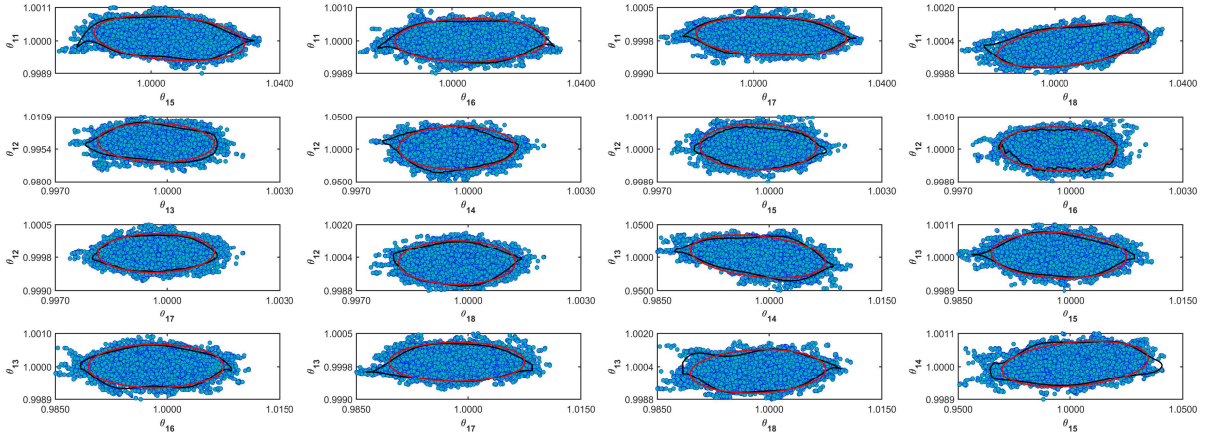


Figure A.9: Coverage regions of parameters 09.  
 (—) Gaussian region; (—) Possolo (2010) region.

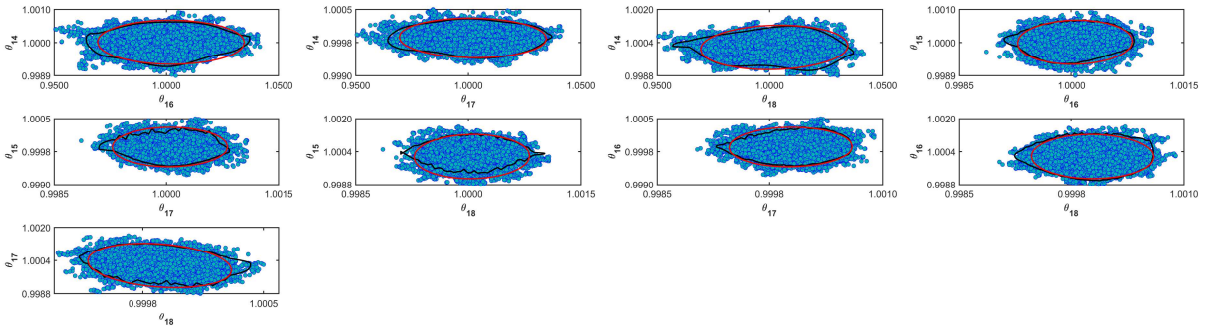


Figure A.10: Coverage regions of parameters 10.  
 (—) Gaussian region; (—) Possolo (2010) region.

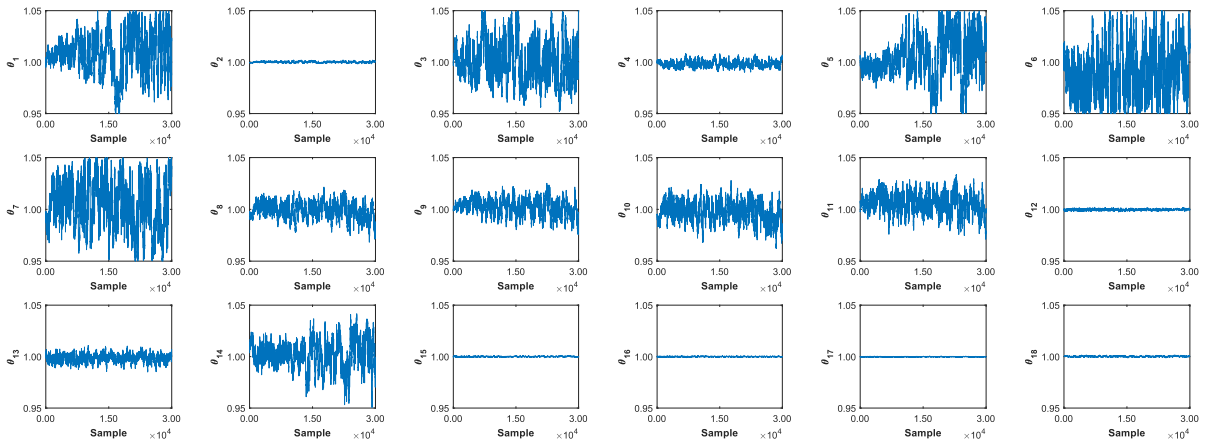


Figure A.11: Parameters random walk.