**UNIVERSITY OF SÃO PAULO
SÃO CARLOS SCHOOL OF ENGINEERING**

Raphael Luiz Vicente Fortulan

**Considering Uncertainties in Power System Transient
Stability Analysis**

São Carlos

2019

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Raphael Luiz Vicente Fortulan

**Considerando Incertezas na Análise de Estabilidade
Transitória em Sistemas Elétricos de Potência**

São Carlos

2019

Raphael Luiz Vicente Fortulan

**Considering Uncertainties in Power System Transient
Stability Analysis**

Dissertation presented to the São Carlos School of Engineering, University of São Paulo, to obtain the degree of Master of Science - Graduate Program in Electrical Engineering.

Concentration area: Eletric Power Systems

Supervisor: Prof. Dr. Luís Fernando Costa
Alberto

Original version

**São Carlos
2019**

I HEREBY AUTHORIZE THE TOTAL OR PARTIAL REPRODUCTION OF THIS WORK,
BY ANY CONVENTIONAL OR ELECTRONIC MEDIUM, FOR RESEARCH PURPOSES,
PROVIDED THAT THE SOURCE IS ACKNOWLEDGED.

Fortulan, Raphael Luiz Vicente
FF745c Considering Uncertainties in Power System Transient Stability Analysis / Raphael Luiz Vicente Fortulan ; supervisor Luís Fernando Costa Alberto. – São Carlos, 2019.

Dissertation (Master) - Graduate Program in Electrical Engineering and Research area in Electric Power Systems – São Carlos School of Engineering, University of São Paulo, 2019.

1. Parametric uncertainties. 2. PEBS. 3. Interval energy function. 4. Robust critical clearing time. 5. Optimization methods. 6. Sensitivity analysis. I. Title.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **RAPHAEL LUIZ VICENTE FORTULAN.**

Título da dissertação: "Considerando incertezas na análise de estabilidade transitória em sistemas elétricos de potência".

Data da defesa: 11/03/2019.

Comissão Julgadora:

Resultado:

Prof. Titular **Luís Fernando Costa Alberto**
(Orientadora)
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Prof. Dr. **Daniel Dotta**
(Universidade Estadual de Campinas/UNICAMP)

APROVADO

Prof. Dr. **Maurício Barbosa de Camargo Salles**
(Escola Politécnica/EP-USP)

APROVADO

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Associado **Ivan Nunes da Silva**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Murilo Araujo Romero**

ACKNOWLEDGEMENTS

To my life-coaches, my grandparents: your whole life's struggle to give my parents a better life, changed mine and inspires me every day. My humble thanks!

My forever interested, encouraging and always enthusiastic Conceição, affectionately known as Conce: she was always keen to know what I was doing and how I was proceeding, although it is likely that she has never grasped what it was all about! I always enjoyed your screams of joy whenever a significant momentous was reached.

I am grateful to my father and mother Sergio and Marisa, who have provided me through moral and emotional support in my life. I am also grateful to my brother Matheus, other family members, and friends who have supported me along the way.

With a very special mention to my uncles Carlos Alberto, Marcos and André; your advices were always the best.

I would also like to thank my dissertation supervisor Prof. Dr. Luís Fernando Costa Alberto of the São Carlos School of Engineering at University of São Paulo. The door to Prof. Alberto office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I am also grateful to the following university staff: Marisa Fortulan—which also happens to be my mother! So, a double thanks!—; and Daniel Manjini for their unfailing support. And finally, last but by no means least, also to everyone in LACOSEP—it was great sharing the laboratory with all of you during these years.

Thanks for all your encouragement!

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

*"ut ager quamvis fertilis sine cultura fructuosus esse non potest, sic sine doctrina animus;
ita est utraque res sine altera debilis."*

*"as a field, though fertile, cannot yield a harvest without cultivation, no more can the
mind without learning; thus each is feeble without the other."*

Marcus Tullius Cicero – Tusculanae Disputationes

ABSTRACT

Fortulan, R. L. V. **Considering Uncertainties in Power System Transient Stability Analysis**. 2019. 122p. Dissertation (Master of Science) - São Carlos School of Engineering, University de São Paulo, São Carlos, 2019.

Alternative renewable energy sources such as solar, wind, and bio energy have brought uncertainties into the power flow of electric power systems. Purely deterministic tools and models are suitable to guarantee the safe operation of these systems. A necessity arises to develop and use methods that deal with uncertainties in the analysis.

Several methodologies have been proposed to assess the transient stability of power systems considering uncertainties. However, most of these techniques are based on the execution of successive simulations, requiring a high computational effort. Hence, they are limited to off-line applications in small systems.

In view of the foregoing, this work proposes the development of two methodologies to assess—in real time—the transient stability of power systems with parametric and operational uncertainties. The first one is an extension of PEBS, which is a direct method for stability assessment. This method, called robust PEBS, employs an interval energy function of the power system to determine a robust estimate critical clearing time. The second one is the use of optimization methods to find a robust estimate critical clearing time. Notably, we employ the Simulated Annealing and Differential Evolution algorithms. Besides developing methods to estimate the critical clearing time, this work also contributes to the analysis of power systems with uncertainties by introducing a technique to reduce the analysis effort. To be specific, a methodology is proposed to identify the most influential parameters for the transient analysis assessment based on a sensitivity analysis of the generator angles with respect to the system parameters.

Keywords: Parametric uncertainties. PEBS. Interval energy function. Robust critical clearing time. Optimization methods. Sensitivity analysis.

RESUMO

Fortulan, R. L. V. **Considerando Incertezas na Análise de Estabilidade Transitória em Sistemas Elétricos de Potência**. 2019. 122p. Dissertation (Master of Science) - São Carlos School of Engineering, University de São Paulo, São Carlos, 2019.

A utilização crescente de fontes alternativas intermitentes de energia como eólica e solar tem trazido incertezas em relação ao nível de geração e ao fluxo de potência em Sistemas Elétricos de Potência (SEP). Assim, a utilização de modelagem e ferramentas puramente determinísticas na análise de estabilidade não são mais suficientes para garantir a operação segura destes sistemas. Torna-se necessário, então, o desenvolvimento e utilização de métodos que incluam essas incertezas nas análises.

Diversas metodologias foram propostas para avaliar a estabilidade transitória de Sistemas Elétricos de Potência considerando incertezas. Porém, grande parte destas técnicas é baseada em execuções de inúmeras simulações e, portanto, exige grande esforços computacionais e são limitadas a aplicações *off-line* e em sistemas elétricos de pequeno porte.

Face ao exposto, propõe-se o desenvolvimento de uma metodologia rápida para a avaliação de estabilidade transitória de SEP's em tempo real considerando incertezas paramétricas. Esta metodologia deverá ser capaz de fornecer rapidamente um resultado robusto sobre a estabilidade. Para tal pretende-se estender a aplicação do métodos direto PEBS para a análise de estabilidade de sistemas elétricos com incertezas.

Além disso, neste trabalho, aplicam-se métodos de otimização para se encontrar o tempo crítico de abertura sem o uso de uma simulação de Monte Carlo e determinam-se os parâmetros que mais influenciam para o estudo de estabilidade transitória utilizando uma análise de sensibilidade dos ângulos dos geradores em relação aos parâmetros do sistema elétrico de potência.

Palavras-chave: Incertezas paramétricas. PEBS. Função energia intervalar. Tempo crítico de abertura robusto. Métodos de otimização. Análise de Sensibilidade.

LIST OF FIGURES

Figure 1 – Stability in 2D	31
Figure 2 – Instability in 2D	32
Figure 3 – Asymptotic Stability in 2D	33
Figure 4 – Exponential Stability in 1D	33
Figure 5 – Illustration of Lyapunov’s Theorem	35
Figure 6 – Equivalent π model of a transmission line	38
Figure 7 – Transformer model	39
Figure 8 – Shunt sign convention	40
Figure 9 – Current and power flows sign convention of a bus α	41
Figure 10 – Synchronous Machine	42
Figure 11 – Schematic description of powers and torques in a synchronous machine	42
Figure 12 – Internal Node Model – n generators and m buses	46
Figure 13 – Interval Vector $X = ([1, 3], [1, 5])^T$	54
Figure 14 – Example Solution	57
Figure 15 – Wrapping Effect Example	60
Figure 16 – Illustration of Krasovskii-LaSalle Invariance Principle	64
Figure 17 – PEBS	70
Figure 18 – Uniform Invariance Principle Sets	75
Figure 19 – Uniform Invariance Principle — Functions $a(x)$, $b(x)$ and $c(x)$	76
Figure 20 – Robust PEBS Method – Graphical Representation	80
Figure 21 – Eggholder Function	89
Figure 22 – Best Solution Evolution - Simulated Annealing	90
Figure 23 – Best Solution Evolution - Differential Evolution	90
Figure 24 – Flowchart - Principal Component Analysis	98
Figure 25 – Flowchart - Sensitivity Matrix Analysis	99
Figure 26 – Single Machine Infinite Bus System	102
Figure 27 – Energy Curves of Robust PEBS – Fault-on Trajectory	102
Figure 28 – IEEE 14 bus system diagram	104

LIST OF TABLES

Table 1 – Mutation Variants	87
Table 2 – Differential Evolution Strategies	88
Table 3 – System Parameters	102
Table 4 – Uncertainties Case 1	105
Table 5 – Case 1 Results	105
Table 6 – Case 1 Execution Time	106
Table 7 – Uncertainties Case 2	106
Table 8 – Case 2 Results	107
Table 9 – Case 2 Execution Time	107
Table 10 – Uncertainties Case 3	107
Table 11 – Case 3 Results	107
Table 12 – Case 3 Execution Time	108
Table 13 – Sensitivity Analysis Methodology Comparison	110

LIST OF ABBREVIATIONS AND ACRONYMS

PEBS	Potential Energy Boundary Surface
COI	Center of Inertia
asep	asymptotic stable equilibrium point
uep	unstable equilibrium point
cuep	controlling unstable equilibrium point
pu	per unit
f	Relative to fault-on period
pf	Relative to the post-fault period
prf	Relative to the pre-fault period
t_{cl}	Clearing Time
t_{cr}	Critical Clearing Time

LIST OF SYMBOLS

$\partial A(x_e)$	boundary of the stability region $A(x_e)$
α	angular acceleration
θ_i	angular deviation of the i -th machine's rotor relative to COI
δ_i	angular deviation of the i -th machine's rotor relative to synchronous reference
$\pi_{\max}$	largest eigenvalue of $\sum_{j=0}^{l-1} \tilde{S}_j^T \tilde{S}_j$
$\rho_{\max}$	principal component
σ	initial mechanical angle
τ_e	electrical torque
τ_m	mechanical torque
$\phi(x_0, t)$	solution for a dynamical system starting at x_0 , for all t
ω	speed deviation in relation to synchronous speed
ω_s	synchronous speed
$\hat{\omega}$	speed deviation in relation to COI
Ω	region of the euclidean space $\mathbb{R}^n$
$A(x_e)$	stability region of x_e
$B_{\alpha\beta}$	susceptance of line $\alpha - \beta$
D_i	damping coefficient of the i -th synchronous generator
E_α	complex electromotive force at bus α
E_{g_i}	complex electromotive force at the i -th synchronous generator α
$G_{\alpha\beta}$	conductance of line $\alpha - \beta$
H	inertia constant of synchronous generator in s
$\mathbb{I}$	set of all compact intervals over the real line
$I_{\alpha\beta}$	complex electric current between buses $\alpha - \beta$

$\mathbb{I}^n$	set of all n -dimensional interval vectors over $\mathbb{R}$
J	moment of inertia
M	inertia constant of synchronous generator in $\text{kg m}^2 \text{s}^{-1}$
M_T	inertia constant of all generators in the system
$M^\top$	transpose matrix of M
$\ M\ _{2,1}$	$L_{2,1}$ norm of matrix M
P_e	electrical power
P_{COI}	COI power
P_α	net real power injection at bus α
P_m	mechanical power
Q_α	net complex power injection at bus α
S	sensitivity matrix
$\tilde{S}$	normalized sensitivity matrix
$\mathbb{R}^n$	n -dimensional euclidean space
t	time
$t_{\alpha\beta}$	turn ratio of transformer between buses $\alpha - \beta$
$u_{\max}$	eigenvector corresponding to $\pi_{\max}$
$ v $	absolute value of v
$\ \vec{v}\ $	norm of vector $\vec{v}$
$V(\cdot)$	energy function
$\mathcal{V}(\cdot)$	interval extension of $V(\cdot)$
$V_k(\cdot)$	kinetic component of V
$V_p(\cdot)$	potential component of V
X'_d	transient reactance of direct axis
$\underline{x}, \bar{x}$	endpoints of an interval X
$\mathbf{Y}$	nodal admittance matrix

Y_{ij}	element $\alpha - \beta$ of $\mathbf{Y}$
$y_{\alpha\beta}$	admittance between buses $\alpha - \beta$
$y_{\alpha\beta}^{sh}$	shunt admittance between buses $\alpha - \beta$
y_{α}^{sh}	shunt admittance connected to bus α
$\mathbf{Y}_{\text{aug}}$	augmented $\mathbf{Y}$ matrix
$\mathbf{Y}_{\text{red}}$	reduced $\mathbf{Y}$ matrix
z^*	conjugate complex of z
$z_{\alpha\beta}$	impedance or resistive losses and leakage reactance of transformer between buses $\alpha - \beta$

CONTENTS

	Introduction	29
1	NONLINEAR SYSTEMS REVIEW	31
1.1	Stability of Autonomous Systems	31
1.2	Lyapunov Stability	34
1.3	Stability Region	35
1.4	Stability Region Characterization	36
2	POWER SYSTEM MODEL FOR TRANSIENT STABILITY ANALYSIS	37
2.1	Power System Static Model	37
2.1.1	Component Modeling	37
2.1.1.1	Transmission Line	37
2.1.1.2	Transformers	38
2.1.1.3	Unified Branch Model	40
2.1.1.4	Shunts	40
2.1.2	Nodal Formulation	41
2.2	Synchronous Machine Model	42
2.3	Internal Node Model for Transient Analysis	45
3	INTERVAL ARITHMETIC	49
3.1	Intervals	49
3.2	Norms and Distance	51
3.3	Function of Intervals	51
3.3.1	Interval Dependency	52
3.3.2	Useful Functions of Intervals	53
3.4	Interval Vectors and Matrices	54
3.4.1	Interval Matrices	55
3.5	Numerical Interval Methods	56
3.5.1	Solve a Linear System	56
3.5.1.1	Gauss-Seidel Method	56
3.5.1.2	Krawczyk Method	57
3.5.2	Solve a Nonlinear System	58
3.5.2.1	Newton's Method	58
3.5.2.2	Nonlinear Krawczyk Method	58
3.5.3	Solve Interval Differential Equations	59

3.5.3.1	Wrapping Effect	59
3.5.3.2	Numerical Interval Simulation	60
4	DIRECT METHODS	63
4.1	Physical Motivational Example	63
4.2	Mathematical Motivation	63
4.3	Direct Methods in Power Systems	64
4.4	PEBS	66
4.4.1	Calculating V_{cr}	69
5	METHODOLOGIES	73
5.1	Robust PEBS	74
5.1.1	Mathematical Background	74
5.1.2	Robust PEBS for a Single Machine Infinity Bus System	76
5.1.2.1	Interval Extension of the Energy Function	76
5.1.2.2	Algorithm	78
5.1.3	Robust PEBS for a Multi-Machine Power System	78
5.1.3.1	Multi-Machine Robust PEBS Implementation	80
5.2	Optimization Methods	81
5.2.1	Simulated Annealing	83
5.2.2	Differential Evolution	86
5.2.3	Testing the Algorithms	88
5.3	Sensitivity	91
5.3.1	Approximating the Derivative	92
5.3.2	PCA - Principal Component Analysis	94
5.3.3	Analysis of the Sensitivity Matrix	95
5.3.4	K-Means	96
5.3.5	Power System Parameter Sensitivity	97
6	RESULTS	101
6.1	Single-machine	101
6.2	Multi-Machine	104
6.2.1	Methodologies Benchmark	105
6.2.1.1	Case 1	105
6.2.1.2	Case 2	106
6.2.1.3	Case 3	107
6.3	Sensitivity Analysis	109
7	CONCLUSION	111

	Bibliography	113
	APPENDIX	119
	APPENDIX A – APPENDIX	121
A.1	Estimating the Number of Monte Carlo Iterations	121

INTRODUCTION

Security analysis is extremely important to guarantee the continuous and safe operation of electric power systems. A power system is considered safe if, for a given initial operating condition, it is able to regain a state of operating equilibrium after being subjected to a physical disturbance, without violating any limits of operation and losing stability [1].

The crescent use of renewable generation sources, such as solar, wind, and wave has reached levels that cannot be more ignored by the grid operator and contribute—given their intrinsically intermittent nature—to uncertainties in the operation of power systems. Developing computational tools for power system security analysis and, in particular, dynamic security analysis suitable for real-time applications in power systems with uncertainties is very challenging. In this work, the problem of security analysis is studied in the transient stability context.

The classical methodology for transient stability analysis is founded on the numerical integration of the differential equations that models a power system. Generally, several simulations of the disturbance-on system and post-disturbance system are performed until an estimate for the critical clearing time—the greatest time interval to clear the disturbance to ensure the stability of the post-disturbance system—is found.

By contrast, in power systems with uncertainties, the classical methodology is combined with the Monte Carlo method to obtain a probability distribution for the critical clearing time. Despite generating complete and precise results, this approach demands a vast amount of information about the system and a large number of simulations for its execution. The increasing complexity and size of the power systems make the Monte Carlo method application restricted and not adequate to rapidly assess the transient stability of an electric power system.

In this work, we propose the use of two methodologies to assess the transient stability of power systems. The first is an extension of the direct method PEBS (potential energy boundary surface) [2] considering parametric uncertainties. In particular, we use the PEBS extension presented in [3] with an interval arithmetic formulation for the energy function. Interval arithmetic enables us to include the uncertainties in the analysis and obtain closed formulas for stability assessment, allowing a faster assessment. Using statistics, a closed formulation would be impossible to achieve. The second methodology is the use of optimization algorithms to make a robust estimate of the critical clearing time also considering parametric uncertainties. We can employ such algorithms by transforming the transient stability analysis of a power system with uncertainties problem into a constrained minimization problem. Both methodologies have advantages and disadvantages. While the first is faster to analyze the transient stability for a larger set of uncertain parameters,

the second will provide a more accurate result—comparable to one obtained through the Monte Carlo method. The two methodologies are faster than the Monte Carlo method, nonetheless.

A further contribution of this work is the development of a methodology to identify the most influential parameters of power systems for transient stability analysis. The methodology employs a sensitivity analysis of the generator angles with respect to the power system parameters with the Principal Component Analysis [4] and the K-Means [5] clustering technique to identify the most influential parameters. This identification is an important step to all methods used to study the transient stability of power systems with uncertainties as all methodologies, except the proposed PEBS method, are affected by the number of uncertain parameters. Being able to consider a smaller set of uncertainties while maintaining a reasonable estimate for the critical clearing time is, therefore, essential.

We divided the dissertation into seven chapters. In chapter 1, we present a brief overview of the nonlinear system theory required to understand PEBS method. In chapter 2, we present a review of the power system model used in transient stability analysis. In chapter 3, we introduce concepts from interval arithmetic. Notably, we present interval operations, interval functions, and numerical methods to solve linear/nonlinear systems of equations and interval differential equations. In chapter 4, we expose the motivation behind the creation of direct methods and review their classical formulation. Chapter 5 is where we develop the robust PEBS method; the optimization problem and algorithm; and the sensitivity analysis. In chapter 6, we test the methodologies in a single machine infinite bus system and a multi-machine system. Finally, in chapter 7, we draw conclusions about the results obtained. In addition, the Appendix presents a method to estimate the necessary number of Monte Carlo simulations that we employed in this work.

1 NONLINEAR SYSTEMS REVIEW

In this chapter, we review some topics in the mathematical theory of nonlinear systems. Knowledgeable readers may skip this chapter or regard it as a quick reference.

1.1 Stability of Autonomous Systems

Consider the following autonomous system given by:

$$\dot{x} = f(x), \quad (1.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is Lipschitz^I. Let $x_e \in \mathbb{R}^n$ be an equilibrium point of (1.1). This way, $f(x_e) = 0$.

Definition 1.1.1 (Flow of Differential Equation [6]) *The flow $\phi(t, x_0)$ is the unique solution of the ODE (1.1) with initial condition $x(t_0) = x_0$.*

Definition 1.1.2 (Stability of Equilibrium Point) *The equilibrium point x_e of (1.1) is stable if, for each $\epsilon > 0$, exists a $\delta = \delta(\epsilon)$ such that $\|x_0 - x_e\| < \delta \Rightarrow \|\phi(t, x_0) - x_e\| < \epsilon$, for all $t \geq t_0$.*

Figure 1 illustrates the concept of stability in two dimensions: The differential equation solution starting at a point inside the ball of radius δ stays inside the ball of radius ϵ .

Figure 1: Stability in 2D

Source: Adapted from Fortulan [7]

^I A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is Lipschitz in $x \in \mathbb{R}^n$ if there is a constant C , such that $\|f(y) - f(x)\| \leq C\|y - x\|$, $\forall y \in \mathbb{R}^n$

Instability, on the other hand, is defined as follows:

Definition 1.1.3 (Instability of Equilibrium Point) *The equilibrium point x_e of (1.1) is unstable if it is not stable.*

Figure 2 illustrates the concept of instability in two dimensions: The differential equation solution starting at a point inside the ball of radius δ escapes from the ball of radius ϵ .

Figure 2: Instability in 2D

Source: Adapted from Fortulan [7]

An equilibrium point can also be asymptotically stable as defined bellow.

Definition 1.1.4 (Asymptotic Stability) *The equilibrium point x_e of (1.1) is asymptotically stable if it is stable and δ can be chosen such that $\|x_0 - x_e\| < \delta \Rightarrow \lim_{t \rightarrow +\infty} \|\phi(t, x_0) - x_e\| = 0$.*

Figure 3 illustrates the concept of asymptotic stability in two dimensions: The differential equation solution starting at a point inside the ball of radius δ converges to the equilibrium point.

Figure 3: Asymptotic Stability in 2D

Source: Adapted from Fortulan [7]

A special case of asymptotically stable is defined as follows:

Definition 1.1.5 (Exponential Stability) *The equilibrium point x_e is exponentially stable if it is asymptotically stable and there exist $\delta > 0$, $\epsilon > 0$ and $\alpha > 0$ such that $\|x_0 - x_e\| < \delta \Rightarrow \|\phi(t, x_0) - x_e\| \leq \epsilon \|x_0 - x_e\| e^{-\alpha(t-t_0)}$, for all $t \geq t_0$.*

Figure 4 illustrates the concept of exponential stability in one dimension: The differential equation solution starting at a point inside the ball of radius ϵ converges exponentially—with a α decay—to the equilibrium point.

Figure 4: Exponential Stability in 1D

Definition 1.1.6 (Hyperbolic Equilibrium Point) *An equilibrium point x_e is hyperbolic if the Jacobian matrix $\mathbf{Df}(x)$ ^{II} at x_e has no eigenvalues on the imaginary axis.*

Definition 1.1.7 (Stable Manifold of Equilibrium Point) *Let x_e be a hyperbolic equilibrium point of (1.1), then the stable manifold of x_e is the set*

$$W^s(x_e) := \left\{ x \in \mathbb{R}^n : \lim_{t \rightarrow +\infty} \phi(t, x) = x_e \right\}. \quad (1.2)$$

Definition 1.1.8 (Unstable Manifold of Equilibrium Point) *Let x_e be a hyperbolic equilibrium point of (1.1), then the unstable manifold of x_e is the set*

$$W^u(x_e) := \left\{ x \in \mathbb{R}^n : \lim_{t \rightarrow -\infty} \phi(t, x) = x_e \right\}. \quad (1.3)$$

1.2 Lyapunov Stability

Lyapunov's theory aims to determine the stability of an equilibrium point of (1.1) without explicitly obtaining the solution $\phi(\cdot)$. For this, it generalizes the concept of energy for a conservative system in physical systems using an energy-like function V . The statement of Lyapunov's Theorem is given below.

Theorem 1.2.1 (Lyapunov's Theorem) *Let $x_e = 0$ be an equilibrium point of (1.1) and $D \subset \mathbb{R}^n$ be an open set containing x_e . Let $V(x) : D \rightarrow \mathbb{R}$ be a continuous differentiable function such that:*

$$V(0) = 0 \text{ and } V(x) > 0, \text{ in } D \setminus \{0\}, \quad (1.4)$$

$$\dot{V}(x) = \sum_{i=1}^n \frac{\partial V}{\partial x_i} f_i(x), \quad (1.5)$$

$$\dot{V} \leq 0 \text{ in } D. \quad (1.6)$$

Then, x_e is stable. Furthermore, if

$$\dot{V}(x) < 0 \text{ in } D \setminus \{0\}, \quad (1.7)$$

then x_e is asymptotically stable.

---

^{II} Let $U \subset \mathbb{R}^n$, $f : U \rightarrow \mathbb{R}^m$. The Jacobian matrix of f is then the matrix

$$\mathbf{Df}(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix},$$

where $f_1, f_2, \dots, f_m$ are the coordinates functions of f and $x_1, x_2, \dots, x_n$ denote the coordinates in $\mathbb{R}^n$ [8].

Figure 5 illustrates the asymptotic stability conclusion of this theorem. Since $\dot{V} < 0$, V must continually diminish along $x(t)$ and $x(t)$ will cut across level sets $V(x) = C$ with ever smaller values of C . In the end, since $V(x) = 0$ only at the equilibrium x_e , $x(t)$ will be trapped at the equilibrium. On the other hand, if $\dot{V} \leq 0$ then $x(t)$ does not necessarily have to cut across level sets of V , since there is no guarantee that V diminishes along $x(t)$.

Figure 5: Illustration of Lyapunov's Theorem

Theorem 1.2.1, however, requires very restricted conditions for V . In fact, to prove stability one needs $\dot{V}(x) < 0$ in $D \setminus \{0\}$. In contrast, the Krasovskii-LaSalle invariance principle^{III}—which we will discuss in chapter 4—enables one to conclude asymptotic stability of an equilibrium point even when $\dot{V} \leq 0$ in D . For the sake of completeness, we introduce the following definition of invariant sets:

Definition 1.2.1 (Invariant set) *The set $M \in \mathbb{R}^n$ is said to be:*

- *an invariant set with respect to (1.1) if $x_0 \in M \Rightarrow \phi(t, x_0) \in M, \forall t \in \mathbb{R}$;*
- *a positively invariant set with respect to (1.1) if $x_0 \in M \Rightarrow \phi(t, x_0) \in M, \forall t \geq 0$.*

1.3 Stability Region

Suppose that x_e is an asymptotically stable equilibrium point of (1.1), then there exists a region $A(x_e)$ containing x_e such that every trajectory initiated in this region converges to the stable equilibrium point x_e as the time goes to infinity. The region $A(x_e)$ is denoted as the stability region of x_e . Formally, $A(x_e)$ is defined as follows:

Definition 1.3.1 (Stability Region) *The stability region of an asymptotically stable equilibrium point x_e is the set of states x_0 such that $\lim_{t \rightarrow +\infty} \phi(t, x_0) = x_e$. Compactly:*

$$A(x_e) := \left\{ x_0 \in \mathbb{R}^n : \lim_{t \rightarrow +\infty} \phi(t, x_0) = x_e \right\}. \quad (1.8)$$

^{III} Also known as the Invariance principle or LaSalle's invariance principle.

The topological boundary of the stability region $A(x_e)$ is denoted by $\partial A(x_e)$.

1.4 Stability Region Characterization

To characterize the stability region, we begin by defining an energy function $V(x)$ as follows:

Definition 1.4.1 (Energy Function) *A function $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is an energy function for (1.1) if it satisfies the following conditions:*

1. $\dot{V}(x(t)) \leq 0$;
2. The set $\{t \in \mathbb{R} : \dot{V}(x(t)) = 0\}$ has measure zero in $\mathbb{R}$, for all non trivial trajectory $x(t)$;
3. If $V(x(t))$ is bounded for $t > t_0$, then $x(t)$ is also bounded for $t > t_0$.

The following theorem describes the behavior of the trajectories of the system at the boundary of the stability region.

Theorem 1.4.1 (Trajectories in the Stability Region Boundary) *If (1.1) admits an energy function, then every trajectory in the boundary of the stability region converges to one of the equilibrium points in the boundary as time goes to infinity.*

Theorem 1.4.1 indicates that every trajectory in the boundary of the stability region has only one behavior: converges to an equilibrium point as time approaches infinity. Therefore, there is no oscillating behavior, neither chaotic nor unbounded trajectories. This theorem also offers a way to characterize the boundary of the stability region. In fact, exploring Theorem 1.4.1, we can affirm that the boundary is composed of unstable hyperbolic equilibrium points, let us say $x_i, i = 1, 2, \dots$, and points whose trajectories converge to x_i as time goes to infinity. Formally, it is the same as saying that the boundary of the stability region is composed of the union of the stable manifolds of x_i . The following corollary summarizes this discussion.

Corollary 1.4.1.1 (Characterization of the Stability Region Boundary) *If there is an energy function for (1.1), then the boundary of the stability region $\partial A(x_e)$ is contained in the set composed of the union of the stable manifolds of the unstable equilibrium points lying in $\partial A(x_e)$, i.e.,*

$$\partial A(x_e) \subseteq \bigcup_{x_i \in \partial A(x_e)} W^s(x_i). \quad (1.9)$$

2 POWER SYSTEM MODEL FOR TRANSIENT STABILITY ANALYSIS

In this chapter, we briefly describe the power system model employed in transient stability analysis. Knowledgeable readers may skip this chapter or regard it as a quick reference.

2.1 Power System Static Model

This section reviews the power flow equations that are used to describe the static behavior of a power system. They are employed in transient stability studies to find both the pre- and post-disturbance conditions.

2.1.1 Component Modeling

2.1.1.1 Transmission Line

A transmission line is modeled by its equivalent π model as presented in Figure 6. In general, three parameters are given: series impedance $z_{\alpha\beta}$; shunt admittances $y_{\alpha\beta}^{sh}$ and $y_{\beta\alpha}^{sh}$ [9]. Line series impedance and corresponding admittance are as follows:

$$z_{\alpha\beta} = r_{\alpha\beta} + jx_{\alpha\beta}, \quad (2.1)$$

$$y_{\alpha\beta} = z_{\alpha\beta}^{-1} = g_{\alpha\beta} + jb_{\alpha\beta}, \quad (2.2)$$

where the series conductance $g_{\alpha\beta}$ and the series susceptance $b_{\alpha\beta}$ are described below:

$$g_{\alpha\beta} = \frac{r_{\alpha\beta}}{r_{\alpha\beta}^2 + x_{\alpha\beta}^2}, \quad (2.3)$$

$$b_{\alpha\beta} = -\frac{x_{\alpha\beta}}{r_{\alpha\beta}^2 + x_{\alpha\beta}^2}. \quad (2.4)$$

Shunt admittance is usually expressed as below:

$$y_{\alpha\beta}^{sh} = g_{\alpha\beta}^{sh} + jb_{\alpha\beta}^{sh}. \quad (2.5)$$

Figure 6: Equivalent π model of a transmission line

The complex voltages of the buses are:

$$E_\alpha = V_\alpha e^{j\theta_\alpha}, \quad (2.6)$$

$$E_\beta = V_\beta e^{j\theta_\beta}. \quad (2.7)$$

Finally, the complex currents $I_{\alpha\beta}$ and $I_{\beta\alpha}$ can be expressed as follows:

$$I_{\alpha\beta} = y_{\alpha\beta} (E_\alpha - E_\beta) + y_{\alpha\beta}^{sh} E_\alpha, \quad (2.8)$$

$$I_{\beta\alpha} = y_{\beta\alpha} (E_\beta - E_\alpha) + y_{\beta\alpha}^{sh} E_\beta. \quad (2.9)$$

The complex power, $S_{\alpha\beta} = P_{\alpha\beta} + jQ_{\alpha\beta}$, is given by

$$S_{\alpha\beta} = E_\alpha I_{\alpha\beta}^* = y_{\alpha\beta}^* V_\alpha e^{j\theta_\alpha} (V_\alpha e^{-j\theta_\alpha} - V_\beta e^{-j\theta_\beta}) - j b_{\alpha\beta}^{sh} V_\alpha^2. \quad (2.10)$$

where the conductance $g_{\alpha\beta}^{sh}$ has been neglected.

Taking the real and imaginary parts of (2.10) to find the expressions for $P_{\alpha\beta}$ and $Q_{\alpha\beta}$ yields:

$$P_{\alpha\beta} = V_\alpha^2 g_{\alpha\beta} - V_\alpha V_\beta g_{\alpha\beta} \cos \theta_{\alpha\beta} - V_\alpha V_\beta b_{\alpha\beta} \sin \theta_{\alpha\beta}, \quad (2.11)$$

$$Q_{\alpha\beta} = -V_\alpha^2 (b_{\alpha\beta} + b_{\alpha\beta}^{sh}) + V_\alpha V_\beta b_{\alpha\beta} \cos \theta_{\alpha\beta} - V_\alpha V_\beta g_{\alpha\beta} \sin \theta_{\alpha\beta}, \quad (2.12)$$

where $\theta_{\alpha\beta} = \theta_\alpha - \theta_\beta$.

2.1.1.2 Transformers

The transformer is modeled by an ideal transformer with turn ratio $t_{\alpha\beta}$ in series with an impedance $z_{\alpha\beta}$ which represents resistive losses and the leakage reactance [9].

Figure 7: Transformer model

Assuming that $t_{\alpha\beta} = a_{\alpha\beta}e^{j\phi_{\alpha\beta}}$, the relation between the complex voltages E_α and E_γ , as shown in Figure 7, is:

$$\frac{E_\gamma}{E_\alpha} = t_{\alpha\beta} = a_{\alpha\beta}e^{j\phi_{\alpha\beta}}. \quad (2.13)$$

Thus, $\theta_\gamma = \theta_\alpha + \phi_{\alpha\beta}$ and $V_\gamma = a_{\alpha\beta}V_\alpha$. As there is no power losses between the buses α and γ , we can write the following expression:

$$E_\alpha I_{\alpha\beta}^* + E_\gamma I_{\beta\alpha}^* = 0. \quad (2.14)$$

Applying (2.13) and (2.14) yields:

$$\frac{I_{\alpha\beta}}{I_{\beta\alpha}} = -t_{\alpha\beta}^* = -a_{\alpha\beta}e^{-j\phi_{\alpha\beta}}. \quad (2.15)$$

Finally, the complex currents $I_{\alpha\beta}$ and $I_{\beta\alpha}$ can be written as follows:

$$I_{\alpha\beta} = -t_{\alpha\beta}^* y_{\alpha\beta} (E_\beta - E_\gamma) = (a_{\alpha\beta}^2 y_{\alpha\beta}) E_\alpha + (-t_{\alpha\beta}^* y_{\alpha\beta}) E_\beta, \quad (2.16)$$

$$I_{\beta\alpha} = y_{\alpha\beta} (E_\beta - E_\gamma) = (-t_{\alpha\beta} y_{\alpha\beta}) E_\alpha + (y_{\alpha\beta}) E_\beta. \quad (2.17)$$

Rewriting (2.16) one obtains that:

$$I_{\alpha\beta} = a_{\alpha\beta} y_{\alpha\beta} e^{-j\phi_{\alpha\beta}} (a_{\alpha\beta} E_\alpha e^{j\phi_{\alpha\beta}} - E_\beta). \quad (2.18)$$

Therefore, the complex power, $S_{\alpha\beta} = P_{\alpha\beta} + jQ_{\alpha\beta}$, is given by:

$$S_{\alpha\beta} = E_\alpha I_{\alpha\beta}^* = y_{\alpha\beta}^* a_{\alpha\beta} V_\alpha e^{j(\theta_\alpha + \phi_{\alpha\beta})} (a_{\alpha\beta} V_\alpha e^{-j(\theta_\alpha + \phi_{\alpha\beta})} - V_\beta e^{-j\theta_\beta}). \quad (2.19)$$

Separating the real and imaginary parts of (2.19) to find the expressions for $P_{\alpha\beta}$ and $Q_{\alpha\beta}$ yields:

$$P_{\alpha\beta} = (a_{\alpha\beta} V_\alpha)^2 g_{\alpha\beta} - (a_{\alpha\beta} V_\alpha) V_\beta g_{\alpha\beta} \cos(\theta_{\alpha\beta} + \phi_{\alpha\beta}) - (a_{\alpha\beta} V_\alpha) V_\beta b_{\alpha\beta} \sin(\theta_{\alpha\beta} + \phi_{\alpha\beta}), \quad (2.20)$$

$$Q_{\alpha\beta} = -(a_{\alpha\beta} V_\alpha)^2 b_{\alpha\beta} + (a_{\alpha\beta} V_\alpha) V_\beta b_{\alpha\beta} \cos(\theta_{\alpha\beta} + \phi_{\alpha\beta}) - (a_{\alpha\beta} V_\alpha) V_\beta g_{\alpha\beta} \sin(\theta_{\alpha\beta} + \phi_{\alpha\beta}). \quad (2.21)$$

2.1.1.3 Unified Branch Model

Combining equations (2.11), (2.12), (2.20) and (2.21) it is possible to find a general expression for the active and reactive powers [9], as seen below:

- From bus α to β

$$P_{\alpha\beta} = (a_{\alpha\beta}V_{\alpha})^2 g_{\alpha\beta} - (a_{\alpha\beta}V_{\alpha}) V_{\beta} g_{\alpha\beta} \cos(\theta_{\alpha\beta} + \phi_{\alpha\beta}) - (a_{\alpha\beta}V_{\alpha}) V_{\beta} b_{\alpha\beta} \sin(\theta_{\alpha\beta} + \phi_{\alpha\beta}), \quad (2.22)$$

$$Q_{\alpha\beta} = -(a_{\alpha\beta}V_{\alpha})^2 (b_{\alpha\beta} + b_{\alpha\beta}^{sh}) + (a_{\alpha\beta}V_{\alpha}) V_{\beta} b_{\alpha\beta} \cos(\theta_{\alpha\beta} + \phi_{\alpha\beta}) - (a_{\alpha\beta}V_{\alpha}) V_{\beta} g_{\alpha\beta} \sin(\theta_{\alpha\beta} + \phi_{\alpha\beta}). \quad (2.23)$$

- From bus β to α

$$P_{\beta\alpha} = V_{\beta}^2 g_{\alpha\beta} - (a_{\alpha\beta}V_{\alpha}) V_{\beta} g_{\alpha\beta} \cos(\theta_{\alpha\beta} - \phi_{\alpha\beta}) - (a_{\alpha\beta}V_{\alpha}) V_{\beta} b_{\alpha\beta} \sin(\theta_{\alpha\beta} - \phi_{\alpha\beta}), \quad (2.24)$$

$$Q_{\beta\alpha} = -V_{\beta}^2 (b_{\alpha\beta} + b_{\alpha\beta}^{sh}) + (a_{\alpha\beta}V_{\alpha}) V_{\beta} b_{\alpha\beta} \cos(\theta_{\alpha\beta} - \phi_{\alpha\beta}) - (a_{\alpha\beta}V_{\alpha}) V_{\beta} g_{\alpha\beta} \sin(\theta_{\alpha\beta} - \phi_{\alpha\beta}). \quad (2.25)$$

2.1.1.4 Shunts

The sign convention used in modeling shunts elements is that the current from a shunt is defined as positive when injected into the bus [9]. Therefore,

$$I_{\alpha}^{sh} = -y_{\alpha}^{sh} E_{\alpha}, \quad (2.26)$$

where E_{α} is the complex voltage at the node α . The injected complex power is as follows:

$$S_{\alpha}^{sh} = P_{\alpha}^{sh} + jQ_{\alpha}^{sh} = E_{\alpha} (I_{\alpha}^{sh})^* = - (y_{\alpha}^{sh})^* V_{\alpha}^2. \quad (2.27)$$

Figure 8 illustrates the sign convention when modeling shunts elements.

Figure 8: Shunt sign convention

2.1.2 Nodal Formulation

The net complex power injection at a network bus (Figure 9) can be calculated using expressions (2.22), (2.23) and (2.27) [9]. Thus, the formulas for the active and reactive powers, at bus α , can be written as follows:

$$P_\alpha = V_\alpha^2 g_\alpha^{sh} + \sum_{\beta \in \Omega_\alpha} P_{\alpha\beta}, \quad (2.28)$$

$$Q_\alpha = V_\alpha^2 b_\alpha^{sh} + \sum_{\beta \in \Omega_\alpha} Q_{\alpha\beta}, \quad (2.29)$$

where Ω_α is the set of adjacent buses to bus α , excluding bus α .

Figure 9: Current and power flows sign convention of a bus α

In practice, equations (2.28) and (2.29) are written using the nodal admittance matrix $\mathbf{Y}$, as constructed below:

$$Y_{\alpha\beta} = -t_{\alpha\beta}^* y_{\alpha\beta}, \quad (2.30)$$

$$Y_{\alpha\alpha} = y_\alpha^{sh} + \sum_{\beta \in \Omega_\alpha} a_{\alpha\beta}^2 (y_{\alpha\beta}^{sh} + y_{\alpha\beta}). \quad (2.31)$$

Considering that $\mathbf{Y} = \mathbf{G} + j\mathbf{B}$, the expressions for active and reactive power injections are:

$$P_\alpha = V_\alpha \sum_{\beta \in \Omega_A} V_\beta (G_{\alpha\beta} \cos \theta_{\alpha\beta} + B_{\alpha\beta} \sin \theta_{\alpha\beta}), \quad (2.32)$$

$$Q_\alpha = V_\alpha \sum_{\beta \in \Omega_A} V_\beta (G_{\alpha\beta} \sin \theta_{\alpha\beta} - B_{\alpha\beta} \cos \theta_{\alpha\beta}), \quad (2.33)$$

where Ω_A is the set of adjacent buses to bus α , including bus α .

2.2 Synchronous Machine Model

In this section, we will derive the swing equation, which is the mathematical relation describing how the rotor of a synchronous machine will swing when there is a mismatch between the mechanical power injected in the machine and the electric power extracted from it [10].

First, consider a synchronous machine as depicted in Figure 10.

Figure 10: Synchronous Machine

Source: Reprinted from Fortulan [7]

Schematically, the different torques and powers of a machine can be illustrated as in Figure 11.

Figure 11: Schematic description of powers and torques in a synchronous machine

Source: Adapted from Fortulan [7]

Applying Newton's Laws of Rotational Motion^I [11] in the generator, the following equations are found:

$$\theta = \frac{S}{r} \quad \text{rad}, \quad (2.34)$$

$$\omega = \frac{d\theta}{dt} \quad \text{rad} \cdot \text{s}^{-1}, \quad (2.35)$$

$$\alpha = \frac{d\omega}{dt} \quad \text{rad} \cdot \text{s}^{-2}. \quad (2.36)$$

The torque due to a tangential force at a distance r from the axis of rotation is described as:

$$\tau = \int r dF \quad \text{N} \cdot \text{m}, \quad (2.37)$$

The force dF , in turn, can be described as the tangential force for a mass particle dm . Thus, the infinitesimal torque $d\tau$ will be the tangential force multiplied by the distance r . The total torque τ will be the sum of all infinitesimal components $d\tau$. This way:

$$dF = adm = \alpha r dm \quad \text{N}, \quad (2.38)$$

$$d\tau = r dF = r^2 \alpha dm \quad \text{N} \cdot \text{m}, \quad (2.39)$$

$$\tau = \int d\tau = \alpha \int r^2 dm \quad \text{N} \cdot \text{m}. \quad (2.40)$$

The moment of inertia of the machine is described as follows:

$$J = \frac{\tau}{\alpha} \Rightarrow J\alpha = \tau \Rightarrow J \frac{d^2\theta_m}{dt^2} = \tau \quad \text{N} \cdot \text{m}^2. \quad (2.41)$$

As shown by equation (2.40), τ represents the total torque applied to the rotor. If we divide τ into an electrical τ_e and a mechanical τ_m component^{II}, we get:

$$\tau = \tau_m - \tau_e. \quad (2.42)$$

Analyzing (2.34) and (2.40), one concludes that θ_m is measured relative to a stationary reference, and therefore grows continuously even in steady-state operation. We correct the problem by defining a mechanical angle δ_m referenced to a rotating axis:

$$\theta_m = t\omega_{sm} + \sigma + \delta_m, \quad (2.43)$$

where ω_{sm} is the synchronous speed of rotation and σ is the initial mechanical angle.

Differentiating two times equation (2.43) with respect to time, we obtain:

$$\frac{d\delta_m}{dt} = \frac{d\theta_m}{dt} - \omega_{sm}, \quad (2.44)$$

^I In the absence of the application of a resulting net torque, the angular velocity remains constant; 2) $\tau = J\alpha$; and 3) For each applied torque, there is a reaction torque with the same magnitude and the opposite direction.

^{II} In generator convention, τ_e slows down the movement, while τ_m speeds it up.

$$\frac{d^2\delta_m}{dt^2} = \frac{d^2\theta_m}{dt^2}. \quad (2.45)$$

Rewriting (2.41) with respect to δ_m , τ_e e τ_m , we get:

$$J \frac{d^2\delta_m}{dt^2} = \tau_m - \tau_e, \quad (2.46)$$

or in power:

$$\omega_m \left(J \frac{d^2\delta_m}{dt^2} \right) = P_m - P_e, \quad (2.47)$$

where P_e and P_m in W are the electrical and mechanical power acting on the rotor, respectively.

Until now we have only dealt with mechanical variables—angle and speed. To convert everything into electrical variables, we recall that for a p -pole machine:

$$\omega_e = \frac{p}{2} \omega_m. \quad (2.48)$$

Differentiating, we obtain:

$$\dot{\omega}_e = \frac{p}{2} \dot{\omega}_m, \quad (2.49)$$

which is:

$$\ddot{\delta}_e = \frac{p}{2} \ddot{\delta}_m. \quad (2.50)$$

Plugging (2.50) into (2.47) yields in:

$$\omega_m \left(J \ddot{\delta}_e \frac{2}{p} \right) = P_m - P_e. \quad (2.51)$$

Another way to write (2.51) is by introducing the inertia constant H :

$$H = \frac{\text{Stored kinect energy at synchronous speed}}{\text{machine rating}} \frac{\text{MJ}}{\text{MVA}} = \frac{\frac{1}{2} J \omega_{sm}^2}{S_B} \text{ s}, \quad (2.52)$$

where S_B is the power rating of the machine in MVA.

Substituting (2.52) into (2.51) we obtain:

$$\omega_m \left(\frac{2S_B H}{\omega_{sm}^2} \ddot{\delta}_e \frac{2}{p} \right) = P_m - P_e. \quad (2.53)$$

In steady-state operation, $\omega_m \approx \omega_{sm}$. Thus, (2.53) can be written as:

$$\frac{2S_B H}{\omega_{sm} \frac{p}{2}} \ddot{\delta}_e = P_m - P_e. \quad (2.54)$$

Lastly, dividing both sides by S_B to convert P_m and P_e into per unit and noting that $\omega_{sm} \frac{p}{2}$ is the synchronous electrical frequency (ω_{se}), we get:

$$\frac{2H}{\omega_{se}} \ddot{\delta}_e = P_m - P_e, \quad (2.55)$$

where P_m and P_e are in per unit.

Assuming a damping proportional to the speed of the machine, we modify (2.55) accordingly:

$$\frac{2H}{\omega_s} \ddot{\delta} = P_m - P_e - D\dot{\delta}, \quad (2.56)$$

where D is the damping constant and, for simplicity, δ and ω_s are, respectively, δ_e and ω_{s_e} . As a remark, equation (2.56) is known as the swing equation [12].

2.3 Internal Node Model for Transient Analysis

In this section, we review the network model for transient stability analysis in power systems. To accomplish this, we make some assumptions:

- Network:
  - The transmission system is in a sinusoidal steady-state.
- Load:
  - The loads connected to the system are constant impedances.
- Synchronous Generator:
  - The machine is represented by a voltage source of constant magnitude in series with a reactance, transient reactance of direct axis);
  - The voltage phase angle coincides with the electric angle of the rotor with respect to the rotating synchronous reference.

Figure 12 illustrates the internal node model with these assumptions.

Considering the network admittance matrix $\mathbf{Y}$ as shown in (2.30) and (2.31), for each generator connect to the grid we need to increase the dimension of $\mathbf{Y}$ by one. Therefore, for each internal bus i connected to a bus k , $\mathbf{Y}$ is modified as follows:

- Add a new column and line for the bus i with $Y_{ii} = -\frac{j}{X'_{d_i}}$ and all non-diagonal elements being zero except $Y_{ik} = Y_{ki} = \frac{j}{X'_{d_i}}$;
- Add $\frac{j}{X'_{d_i}}$ to the k -th diagonal element of $\mathbf{Y}$: $Y_{kk} + \frac{j}{X'_{d_i}}$,

in which X'_{d_i} is the i -th machine transient reactance of direct axis.

At the end of the iterative process described above, we obtain:

$$\mathbf{Y}_{\text{aug}} = \begin{bmatrix} \mathbf{Y}_g & \mathbf{Y}_1 \\ \mathbf{Y}_2 & \mathbf{Y} \end{bmatrix} \begin{matrix} n \\ m \end{matrix}, \quad (2.57)$$

Figure 12: Internal Node Model – n generators and m buses

where $\mathbf{Y}_{\text{aug}}$ is the augmented admittance matrix; $\mathbf{Y}_g$ is a n -by- n identity matrix containing the transient admittance of all n generators; $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are, respectively, a m -by- n and a n -by- m matrices representing the connection between the internal generator buses and the network; lastly, $\mathbf{Y}$ is the m -by- m network admittance matrix.

From our assumptions, the load connected to the bus i is treated as a constant impedance. Thus,

$$Y_{L_i} = \frac{P_{L_i} - jQ_{L_i}}{V_i^2}, \quad (2.58)$$

where Y_{L_i} is the equivalent admittance; P_{L_i} and Q_{L_i} are, respectively, the real and complex power of the load; and V_i is the voltage at bus i .

Using (2.27), we then modify $\mathbf{Y}_{\text{aug}}$ by subtracting Y_{L_i} from $Y_{\text{aug}_{ii}}$.

$\mathbf{Y}_{\text{aug}}$ is now an augmented matrix containing all loads of the electric power system.

Applying Kirchhoff's first law to each of the power system node, we have:

$$\mathbf{I} = \mathbf{Y}_{\text{aug}} \begin{bmatrix} \mathbf{E} \\ \mathbf{V} \end{bmatrix}, \quad (2.59)$$

where $\mathbf{V}$ and $\mathbf{E}$ are, respectively, the buses and generators voltage vectors; and $\mathbf{I}$ is the nodal injection current vector.

As we only have current injection at the generator buses, it is possible to partition the

vector $\mathbf{I}$ as follows:

$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \\ - \\ 0 \end{bmatrix}, \quad (2.60)$$

where $I_1, I_2, \dots, I_n$ are the complex injected generator currents at the internal generator buses.

Since there are buses with no current injection, we can apply Kron's Reduction [13] to $\mathbf{Y}_{\text{aug}}$ and find a reduced matrix $\mathbf{Y}_{\text{red}}$. Modifying (2.60), we obtain the following equation:

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \mathbf{Y}_{\text{red}} \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}, \quad (2.61)$$

where $E_1, E_2, \dots, E_n$ are the complex voltages at the generator buses.

Finally, applying (2.61) into (2.32), we obtain that the real power injected by the i -th machine into the network is as follows:

$$P_i = |E_i|^2 G_{ii} + \sum_{\substack{j=1 \\ \neq i}}^n |E_i| |E_j| G_{ij} \cos \delta_{ij} + |E_i| |E_j| B_{ij} \sin \delta_{ij}, \quad (2.62)$$

where $\mathbf{G} = \Re\{\mathbf{Y}_{\text{red}}\}$ and $\mathbf{B} = \Im\{\mathbf{Y}_{\text{red}}\}$.

For simplicity sake, we define:

$$|E_i| |E_j| G_{ij} := D_{ij}, \quad (2.63)$$

$$|E_i| |E_j| B_{ij} := C_{ij}. \quad (2.64)$$

Thus,

$$P_i = |E_i|^2 G_{ii} + \sum_{\substack{j=1 \\ \neq i}}^n C_{ij} \sin \delta_{ij} + D_{ij} \cos \delta_{ij}. \quad (2.65)$$

3 INTERVAL ARITHMETIC

This chapter introduces the formulation of algebra with intervals and the numerical methods employed when solving systems of interval equation and interval differential equations. Interval arithmetic plays a key role in this work because all the uncertainties are modeled as intervals.

The greatest advantage of using intervals instead of statistics is that the former does not require iterative procedures for evaluating expressions. There is no need of multiple iterations to verify the probability distribution of a function applied in an interval, as it will be an interval and can be evaluated analytically. In addition, intervals provide rigorous bounds for all calculations, not only mathematically but also computationally. The reason is that when we execute an interval operation on a code, the interpreter/compiler takes into consideration the error introduced by using floating-point numbers.

3.1 Intervals

We begin by defining our notation: $\mathbb{I}$ denotes the set of all compact intervals over the real line of the form $X = [\underline{x}, \bar{x}]$.

Then we define an interval as follows:

Definition 3.1.1 (Interval) *An interval is the set of real numbers given by*

$$X = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \bar{x}\}, \quad (3.1)$$

where $\underline{x}$ and $\bar{x}$ are respectively the left and right endpoints of X .

Intervals also include real numbers as can be seen in the following definition:

Definition 3.1.2 (Degenerate Intervals) *An interval X is called degenerate if $x = \underline{x} = \bar{x}$. In this case, the interval contains only the real number x .*

Now, we define the properties of an algebraic system $\langle \mathbb{I}, +, -, \cdot, / \rangle$ in the set of intervals:

Definition 3.1.3 (Sum of Intervals) *The sum of two intervals X and Y is given by:*

$$X + Y = [\underline{x} + \underline{y}, \bar{x} + \bar{y}] \quad (3.2)$$

This operation is usually called Minkowski sum.

Definition 3.1.4 (Difference of Intervals) *The difference of two intervals X and Y is given by:*

$$X - Y = [\underline{x} - \bar{y}, \bar{x} - \underline{y}] \quad (3.3)$$

This operation is usually called Minkowski difference.

Definition 3.1.5 (Multiplication of Intervals) The product of two intervals X and Y is given by:

$$X \cdot Y = [\min S, \max S], \quad (3.4)$$

where $S = \{\underline{x} \cdot \underline{y}, \underline{x} \cdot \bar{y}, \bar{x} \cdot \underline{y}, \bar{x} \cdot \bar{y}\}$.

Definition 3.1.6 (Division of Intervals) The division of two intervals X and Y , with $0 \notin Y$, is given by:

$$X/Y = [\min S, \max S], \quad (3.5)$$

where $S = \{\underline{x} \cdot 1/\bar{y}, \underline{x} \cdot 1/\underline{y}, \bar{x} \cdot 1/\bar{y}, \bar{x} \cdot 1/\underline{y}\}$.

Despite the operations are defined by the endpoints, they operate binarily. This can be seen below:

$$X \diamond Y = \{x \diamond y : x \in X, y \in Y\}, \quad (3.6)$$

where $\diamond$ is a generic operation.

The relations between intervals are given as follows:

Definition 3.1.7 (Intersection of Intervals) The intersection between two intervals X and Y is:

$$X \cap Y = \begin{cases} \emptyset & \text{if } \bar{y} < \underline{x} \text{ or } \bar{x} < \underline{y}, \\ [\max\{\underline{x}, \underline{y}\}, \min\{\bar{x}, \bar{y}\}] & \text{otherwise.} \end{cases} \quad (3.7)$$

Definition 3.1.8 (Union of Intervals) The union between two intervals X and Y is:

$$X \cup Y = \{z : z \in X \text{ or } z \in Y\} \quad (3.8)$$

In particular, if $X \cap Y \neq \emptyset$, then

$$X \cup Y = [\min\{\underline{x}, \underline{y}\}, \max\{\bar{x}, \bar{y}\}]. \quad (3.9)$$

Definition 3.1.9 (Interval Hull) The interval hull of two intervals X and Y is defined as

$$X \sqcup Y = [\min\{\underline{x}, \underline{y}\}, \max\{\bar{x}, \bar{y}\}]. \quad (3.10)$$

3.2 Norms and Distance

A norm in $\langle \mathbb{I}, +, -, \cdot, / \rangle$ can be given by [14]:

$$\|X\| = \max\{|\underline{x}|, |\bar{x}|\}, X \in \mathbb{I}. \quad (3.11)$$

The distance between two intervals X and Y is given as follows:

Definition 3.2.1 (Distance of Intervals)

$$d(X, Y) = \max\{|\underline{x} - \underline{y}|, |\bar{x} - \bar{y}|\} \quad (3.12)$$

3.3 Function of Intervals

Given a real function f of the variable x , we would like to know precisely the range of values given by $f(x)$ as x sweeps over the interval X . In other words, we would like to know the image of the set X under the mapping f :

$$f(X) = \{f(x) : x \in X\} \quad (3.13)$$

Mathematically it is possible to rewrite (3.13) as follows:

Definition 3.3.1 (Function of an Interval)

$$f(X) = \bigcup_{x \in X} \{f(x)\} \quad (3.14)$$

Clearly, it is impossible to use functions of intervals in practice. We would need to evaluate f over countless points to obtain a precise estimate of its image.

An idea to overcome this problem is to extend f by applying the interval arguments directly in its expression. This is called an interval extension of f . A formal definition is given below:

Definition 3.3.2 (Interval Extension) *We say that F is an interval extension of f if, for a degenerate interval as an argument for F , F coincides with f , i.e., $F([x, x]) = f(x)$.*

Furthermore, we can also define an interesting property of interval extensions called inclusion isotonicity.

Definition 3.3.3 (Inclusion Isotonicity) *An interval extension of F is inclusion isotonic if for every pair of intervals, $Y \subseteq X$, implies $F(Y) \subseteq F(X)$.*

The following theorem of interval analysis due to Moore [15] is remarkable. It guarantees that the values of a real function operating over an interval can be rigorously bounded by its interval extension.

Theorem 3.3.1 (Fundamental Theorem of Interval Analysis) *Let F be an inclusion isotonic interval extension of the real function f . Then,*

$$f(X) \subseteq F(X). \quad (3.15)$$

Until now, nothing has been said about how to construct an interval extension of f . A natural way to build it is simply replacing each real variable in f expression, by an interval variable. This is known as the natural interval extension of f and its formal definition [16] is given below:

Definition 3.3.4 (Natural Interval Extension) *The natural interval extension F is obtained by replacing in the expression of f : each real variable x_i with an interval variable X_i ; each constant k with a degenerate interval $[k, k]$ and each real operation by its corresponding interval operation.*

A natural interval extension has the following property:

Theorem 3.3.2 (Property of Natural Interval Extensions [15]) *The natural interval extension of a real function f is an inclusion isotonic interval extension.*

We can also obtain an interval extension using Taylor series [17]. For this, let f be an $n + 1$ differentiable function, then the interval extension by Taylor series (order n) of f is given as follows:

$$F(X) = f(y) + \sum_{i=1}^n \frac{1}{i!} D^i f(y) \cdot (X - y)^i + R_n, \quad (3.16)$$

where $y \in X$, $D^i f(y)$ is the i -th derivative of f at y and R_n is the interval extension of Lagrange remainder

$$r_n(x, \xi, y) = \frac{1}{(n+1)!} D^{n+1} f(\xi) \cdot (x - y)^{n+1}, \quad (3.17)$$

in which ξ is between x and y .

Usually, the remainder is evaluated by its natural interval extension, y is chosen to be $\frac{\underline{x} + \bar{x}}{2}$ and $\xi = X$.

3.3.1 Interval Dependency

A problem in evaluating an interval expression using interval arithmetic is that each occurrence of the same variable is treated as a new one. This way, the dependency between different occurrences of a variable is lost. The result is an overestimate of f image, as can be seen in the following example:

Example 3.3.1 Consider $f(x) = x^3 - 12x^2 + 48x - 65$ and $X = [1, 2]$. To estimate $f(X)$ we can use the natural interval extension of f by rewriting $f(x)$ as

$$f(x) = x \cdot x \cdot x - 12 \cdot x \cdot x + 48x - 65.$$

$F(X)$ is then

$$F(X) = X \cdot X \cdot X - 12 \cdot X \cdot X + 48X - 65,$$

and $F([1, 2]) = [-64, 27]$. On the other hand, evaluating the natural interval extension of an equivalent formulation in real arithmetic $f(x) = (x - 4)^3 - 1$ yields the exact image $[-28, -9]$.

Using Taylor series, on the other hand, yields a different result. This is shown in the example below:

Example 3.3.2 Consider again the same function $f(x) = x^3 - 12x^2 + 48x - 65$ of Example 3.3.1 and $X = [1, 2]$. Choosing $n = 1$, the interval extension using Taylor series of f is given as follows:

$$F(X) = [y^3 - 12y^2 + 48y - 65] + [3x^2 - 24x + 48] \cdot (X - y) + \underbrace{\left[\frac{1}{2} \cdot (6X - 24) \cdot (X - y)^2 \right]}_{R_1} \quad (3.18)$$

Evaluating R_1 at $y = (1 + 2)/2$ results the interval $[-28.25, -7.25]$. Consequently, we obtained a sharper enclosure than that achieved using a natural interval extension in Example 3.3.1.

In practice, this problem is lessened as several packages for interval arithmetic have implemented routines for the accurate evaluation of common functions like polynomials, trigonometric and logarithmic.

3.3.2 Useful Functions of Intervals

In this subsection, we present some useful measures of intervals and their notations [18].

The width of an interval X is:

$$w(X) = \bar{x} - \underline{x}. \quad (3.19)$$

The midpoint of X is:

$$m(X) = \frac{\underline{x} + \bar{x}}{2}. \quad (3.20)$$

3.4 Interval Vectors and Matrices

In this section, we introduce the definitions and properties of interval vectors and matrices. As such, we begin by reviewing the notation for interval vectors.

$\mathbb{I}^n$ represents the set of all n -dimensional interval vectors of the form $([\underline{x}_1, \bar{x}_1], [\underline{x}_2, \bar{x}_2], \dots, [\underline{x}_n, \bar{x}_n])^\top$ [19]. We understand an interval vector $X \in \mathbb{I}^n$ as the set of all vectors $x \in \mathbb{R}^n$ with $x_i \in [\underline{x}_i, \bar{x}_i]$ for $i = 1, \dots, n$.

Literature, in general, refers to an interval vector as a rectangular box or simply box. This comes from a geometric interpretation of the interval vector as a region in n -space. For example, if $n = 2$, $\mathbb{I}^2$ is a rectangle in the plane (as seen in Figure 13).

Figure 13: Interval Vector $X = ([1, 3], [1, 5])^\top$

We operate interval vectors component-wise, i.e., for $X, Y \in \mathbb{I}^n$:

$$X \diamond Y := X_i \diamond Y_i \quad \text{for } i = 1, \dots, n, \quad (3.21)$$

where $\diamond = \{+, -, \cdot, \setminus\}$ [19].

Likewise, the midpoint of an interval vector is also defined component-wise [19], i.e.,

$$m(X) = (m(X_1), m(X_2), \dots, m(X_n))^\top. \quad (3.22)$$

On the other hand, the norm and the width of an interval vector are given as follows [17]:

$$\|X\| = \max_i \|X_i\|, \quad (3.23)$$

$$w(X) = \max_i w(X_i). \quad (3.24)$$

The intersection of two interval vectors is empty if any intersection of their corresponding components is empty [17]. Otherwise,

$$X \cap Y = (X_1 \cap Y_1, X_2 \cap Y_2, \dots, X_n \cap Y_n)^\top. \quad (3.25)$$

3.4.1 Interval Matrices

Now we are interested in showing the properties and definitions of intervals for matrices.

An m -by- n interval matrix is a rectangular array of intervals [19]:

$$A = (A_{ij}) = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}, \quad (3.26)$$

where $A_{ij} \in \mathbb{I}$.

The set of all m -by- n interval matrices is denoted by $\mathbb{I}^{m \times n}$. We understand an interval matrix as a set of real m -by- n matrices:

$$A = \{\check{A} \in \mathbb{R}^{m \times n} : \check{A}_{ij} \in A_{ij} \text{ for } i = 1, \dots, m; j = 1, \dots, n\}. \quad (3.27)$$

As with real matrices, an 1-by-1 interval matrix is a “scalar” vector and an n -by-1 is an interval vector. We can, therefore, relate some vector properties to matrices, namely:

$$w(A) = \max_{i,j} w(A_{ij}), \quad (3.28)$$

$$m(A) = m((A_{ij})) = \begin{bmatrix} m(A_{11}) & \cdots & m(A_{1n}) \\ \vdots & \ddots & \vdots \\ m(A_{m1}) & \cdots & m(A_{mn}) \end{bmatrix}. \quad (3.29)$$

We define the matrix norm as [17]:

$$\|A\| = \max_i \sum_j \|A_{ij}\|, \quad (3.30)$$

The extension of the sum and difference operations to interval matrices $A, B \in \mathbb{I}^{m \times n}$ are as follows [19]:

$$A \pm B = \{\check{A} \pm \check{B} : \check{A} \in A, \check{B} \in B\}, \quad (3.31)$$

where $A \pm B \in \mathbb{I}^{m \times n}$.

Analogously, the multiplication of $A \in \mathbb{I}^{m \times n}$ and $B \in \mathbb{I}^{n \times p}$ is defined as follows [19]:

$$AB = \{\check{A}\check{B} : \check{A} \in A, \check{B} \in B\}, \quad (3.32)$$

where $AB \in \mathbb{I}^{m \times p}$.

We point out that the interval extension of the real matrix multiplication $C = AB$, where

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}, \quad \text{for } i = 1, \dots, m; j = 1, \dots, p, \quad (3.33)$$

is a sharp enclosure for the multiplication of two interval matrices. It may contain, however, real matrices that are not result of the multiplication of real matrices $\check{A} \in A$ and $\check{B} \in B$ [17].

3.5 Numerical Interval Methods

In this section, we introduce interval numerical methods to solve both linear/non-linear systems of equations and interval differential equations. Their solution plays an instrumental role in this work: the power flow equations are nonlinear and the generator is modeled by a set of dynamical equations.

As a remark, despite linear equations not showing immediately in the transient stability analysis formulation, when solving nonlinear system they appear as an intermediate step. In the following subsection, we introduce two methods for solving a linear system of equations.

3.5.1 Solve a Linear System

In this subsection, we consider a linear system of interval equations represented for example as:

$$AX = B, \quad (3.34)$$

where A is an n -by- n interval matrix, $X \in \mathbb{I}^n$ and $B \in \mathbb{I}^n$. We define the solution set of (3.34) as follows [19]:

$$\varsigma(A, B) := \{\check{x} \in \mathbb{R}^n : \check{A}\check{x} = \check{B} \text{ for } \check{A} \in A, \check{B} \in B\}. \quad (3.35)$$

The set $\varsigma(A, B)$ is not always an interval vector. For example, consider the following interval linear system:

$$\begin{bmatrix} [1, 2] & [4, 5] \\ [7, 8] & [3, 4] \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} [-1, 1] \\ [-1, 1] \end{bmatrix}. \quad (3.36)$$

The solution set of (3.36) is not an interval as is shown in the following figure.

However, we can enclose the solution set with an interval vector as seen in Figure 14.

Some approaches for solving interval linear systems use an interval extension of well-known methods such as Gauss elimination and Gauss-Seidel method [19], others are based in the interval arithmetic theory, e.g., Krawczyk method [20].

In the rest of the section, we will show the methods used in this work. We will not, however, enter into the mathematical details and proofs of each method. The interested reader is referred to [17], [19] for further information.

3.5.1.1 Gauss-Seidel Method

The algorithm for executing the Gauss-Seidel method is as follows:

1. Multiply both sides of (3.34) by a matrix Y :

$$AX = B \rightarrow \underbrace{YA}_G X = \underbrace{YB}_C, \quad (3.37)$$

we usually choose Y to be the inverse of the midpoint matrix of A ;

Figure 14: Example Solution

2. Choose an initial solution X^0 ;
3. Evaluate the sequences:

$$\tilde{X}_i^{k+1} = \frac{1}{G_{ii}} \left(C_i - \sum_{j=1}^{i-1} G_{ij} X_j^{k+1} - \sum_{j=i+1}^n G_{ij} X_j^k \right) \quad i = 1, 2, \dots, n, \quad (3.38)$$

$$X_i^{k+1} = \tilde{X}_i^{k+1} \cap X_i^k \quad i = 1, 2, \dots, n, \quad (3.39)$$

until the solution enclosure is sharp enough^I.

3.5.1.2 Krawczyk Method

The algorithm for executing the Krawczyk method is as follows:

1. Multiply both sides of (3.34) by a matrix Y . Similarity to Gauss-Seidel method, we usually choose $Y = m(A)^{-1}$;
2. Define the matrix $E = I - YA$, where I is the n -by- n identity matrix;
3. Evaluate $\|E\|$. If $\|E\| < 1$ then proceed to the next step. If $\|E\| > 1$ then choose another Y matrix and return to the first step;

^I We verify the sharpness of a solution enclosure examining by how much the enclosure width has changed from one iteration to the next.

4. Evaluate the sequence:

$$X^{k+1} = \{YB + EX^k\} \cap X^k \quad k = 0, 1, 2, \dots, \quad (3.40)$$

with

$$X_i^0 = \frac{[-1, 1]\|YB\|}{1 - \|E\|} \quad i = 1, 2, \dots, n, \quad (3.41)$$

until the solution X^k is sharp enough.

3.5.2 Solve a Nonlinear System

In this subsection, we introduce two methods to solve a system of nonlinear interval equations. For this, consider the following nonlinear vector equation:

$$F(x) = 0, \quad (3.42)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $x \in \mathbb{R}^n$.

3.5.2.1 Newton's Method

Newton's method [21] states that the solution of (3.42) is given by:

$$F'(x^k)(x^{k+1} - x^k) = -F(x^k) \quad k = 1, 2, \dots, \quad (3.43)$$

where $F'(x)$ is the Jacobian matrix of $F(x)$. In parallel, let $\mathcal{F}'(X)$ be the interval extension of $F'(x)$. Then, the interval extension of Newton's method as given in [19] is:

$$\mathcal{F}'(X^k)V = -F(y) \quad k = 1, 2, \dots, \quad (3.44)$$

$$X^{k+1} = X^k \cap \{y + V\} \quad k = 1, 2, \dots, \quad (3.45)$$

where $y \in X^k$. Usually we choose y to be the midpoint of X^k .

3.5.2.2 Nonlinear Krawczyk Method

We now present a nonlinear version of the Krawczyk method. The algorithm for its execution as presented in [17] is as follows:

$$X^{k+1} = X^k \cap K(X^k), \quad (3.46)$$

where

$$K(X^k) = y^k - Y^k F(y^k) + (I - Y^k \mathcal{F}'(X^k))(X^k - y^k), \quad (3.47)$$

$$y^k = m(X^k), \quad (3.48)$$

and

$$Y^k = \begin{cases} [m(\mathcal{F}'(X^k))]^{-1} & \text{if } \|I - [m(F'(X^k))]^{-1} \mathcal{F}'(X^k)\| \leq \|I - Y^{k-1} \mathcal{F}'(X^{k-1})\|, \\ Y^{k-1} & \text{otherwise.} \end{cases} \quad (3.49)$$

K is known in literature as the Krawczyk operator [17].

Further details of the method can be found in [19].

3.5.3 Solve Interval Differential Equations

An interval differential equation is defined as:

$$\dot{X} = f(X), \quad (3.50)$$

where $X \in \mathbb{I}^n$ and $F : \mathbb{I}^n \rightarrow \mathbb{I}^n$. Solving (3.50) is not trivial. While in conventional differential equations, given the solution x_k at a time t_k , we find the solution x_{k+1} at a time t_{k+1} and assume a linear interpolation between the two points, in interval differential equations we must find a box that encloses the interval solutions $[\underline{x}_k, \bar{x}_k]$, $[\underline{x}_{k+1}, \bar{x}_{k+1}]$ and the time interval $[t_k, t_{k+1}]$.

We may now ask how to numerically solve an interval differential equation? A reasonable answer would be “Use an interval extension of numerical methods employed to solve ODEs such as Runge-Kutta, Bogacki-Shampine, etc”. Unfortunately, there are two main problems with this approach: the interval dependency and the "wrapping effect".

As the interval dependency problem was already explored in subsection 3.3.1, we will now discuss the wrapping effect.

3.5.3.1 Wrapping Effect

Let $\mathbb{M}$ be a discrete map, $\mathbb{M} : \mathbb{R}^n \rightarrow \mathbb{R}^n$. This function can be used to discretize the solution of a differential equation in time. So, we can generate a sequence of points $\{x_i\}_{i=0,\dots}$ given an initial point $x_0 \in \mathbb{R}^n$ as follows:

$$x_{i+1} = \mathbb{M}(x_i), \quad \text{for } i = 0, \dots \quad (3.51)$$

Trying to enclose the points in (3.51) using interval arithmetic is not simply [22]. Whereas $\mathbb{M}(X_i)$ is not generally an interval vector, the interval extension of $\mathbb{M}$ evaluated at X_i is an interval vector which encloses, or “wraps”, $\mathbb{M}(X_i)$. Thus, it contains points that are not in the set .

The problem gets even worse if we use an interval iterative procedure:

$$X_{i+1} = \mathcal{M}(X_i), \quad \text{for } i = 0, \dots, \quad (3.52)$$

where $\mathcal{M}$ is an interval extension of $\mathbb{M}$, as erroneous points generate other erroneous points during the whole iterative process. This effect is known in literature as the “wrapping effect” [17].

The combination of the interval dependency and the wrapping effect generates an increasing overestimate of the solution at each time step, illustrated in Figure 15. Several attempts were made to minimize both the interval dependency and wrapping effect problems. In [17], for example, the Taylor expansion of $x(t)$ is used to find an approximation for $x(t+h)$, in which h is the integration step. Literature refers to it as the "Guaranteed integration with Taylor method". We note that good results are only obtained if the Taylor expansion is

Figure 15: Wrapping Effect Example

of high order, making it infeasible to larger systems. Another popular method is to use the Picard–Lindelöf operator^{II} to obtain an estimate of the box enclosing $x(t + h)$ [24]. Finally, we cite the method proposed in [25], which combines a Runge-Kutta method with the Picard–Lindelöf operator to obtain better solutions estimates. Unfortunately, they all have the disadvantage of being too costly to implement for large-scale systems. In the following subsection, we will show a method—employed in this work—that can be applied in larger systems with reasonable performance and accuracy.

3.5.3.2 Numerical Interval Simulation

To minimize the error introduced by the wrapping effect and the interval dependency, we employed the NIS (Numerical Interval Simulation) method proposed in [26], as it was the methodology that provided the best combination of computational efficiency and precision in our tests. NIS performs numerical simulation, using extreme values for the

^{II} Iteration function from the Picard iteration [23].

state variables derivatives. Such extreme values for the derivatives are calculated in the following manner:

1. First, NIS uses the interval extension of f to calculate the interval of possible values for the state variables derivatives;
2. Second, NIS takes the maximum or minimum of the resulting interval, depending on whether the maximal or minimal derivative is required.

The Numerical Interval Simulation method calculates maximal and minimal values for the derivatives at an instant t_k . Such extreme values are used in a numerical routine to determine the state variables value at the next instant t_{k+1} . Several different routines could be used to perform the numerical simulation. In this work, we employed a Runge-Kutta like routine as presented in [26]. The algorithm for executing the method is as follows:

```

while  $t < T$ :
  for  $i$  in  $\text{range}(n)$ :
     $k_{i1 \max} = h \cdot \max\{f_i(t, X_1, X_2, \dots, [\bar{x}_i, \bar{x}_i], \dots, X_n)\}$ ;
     $k_{i1 \min} = h \cdot \min\{f_i(t, X_1, X_2, \dots, [\underline{x}_i, \underline{x}_i], \dots, X_n)\}$ ;
     $k_{i2 \max} = h \cdot \max\{f_i(t + h/2, X_1 + 1/2 \cdot [k_{11 \min}, k_{11 \max}], \dots, [\bar{x}_i + 1/2 \cdot$ 
       $k_{i1 \max}, \bar{x}_i + 1/2 \cdot k_{i1 \max}], \dots, X_n + 1/2 \cdot [k_{n1 \min}, k_{n1 \max}])\}$ ;
     $k_{i2 \min} = h \cdot \max\{f_i(t + h/2, X_1 + 1/2 \cdot [k_{11 \min}, k_{11 \max}], \dots, [\underline{x}_i + 1/2 \cdot$ 
       $k_{i1 \min}, \underline{x}_i + 1/2 \cdot k_{i1 \min}], \dots, X_n + 1/2 \cdot [k_{n1 \min}, k_{n1 \max}])\}$ ;
     $k_{i3 \max} = h \cdot \max\{f_i(t + h/2, X_1 + 1/2 \cdot [k_{12 \min}, k_{12 \max}], \dots, [\bar{x}_i + 1/2 \cdot$ 
       $k_{i2 \max}, \bar{x}_i + 1/2 \cdot k_{i2 \max}], \dots, X_n + 1/2 \cdot [k_{n2 \min}, k_{n2 \max}])\}$ ;
     $k_{i3 \min} = h \cdot \min\{f_i(t + h/2, X_1 + 1/2 \cdot [k_{12 \min}, k_{12 \max}], \dots, [\underline{x}_i + 1/2 \cdot$ 
       $k_{i2 \min}, \underline{x}_i + 1/2 \cdot k_{i2 \min}], \dots, X_n + 1/2 \cdot [k_{n2 \min}, k_{n2 \max}])\}$ ;
     $k_{i4 \max} = h \cdot \max\{f_i(t + h, X_1 + [k_{13 \min}, k_{13 \max}], \dots, [\bar{x}_i + k_{i3 \max}, \bar{x}_i +$ 
       $k_{i3 \max}], \dots, X_n + [k_{n3 \min}, k_{n3 \max}])\}$ ;
     $k_{i4 \min} = h \cdot \min\{f_i(t + h, X_1 + [k_{13 \min}, k_{13 \max}], \dots, [\underline{x}_i + k_{i3 \min}, \underline{x}_i +$ 
       $k_{i3 \min}], \dots, X_n + [k_{n3 \min}, k_{n3 \max}])\}$ ;
     $x_{i \max} = \frac{k_{i1 \max} + 2k_{i2 \max} + 2k_{i3 \max} + k_{i4 \max}}{6}$ ;
     $x_{i \min} = \frac{k_{i1 \min} + 2k_{i2 \min} + 2k_{i3 \min} + k_{i4 \min}}{6}$ ;
     $X_i = [x_{i \min}, x_{i \max}]$ ;
   $t = t + h$ ;

```

Algorithm 1: Numerical Interval Simulation

The interested reader is referred to [17], [22] for other methods and [27] to available packages and libraries.

4 DIRECT METHODS

In this chapter, we discuss direct methods for transient stability analysis of electric power systems. In transient stability analysis, our focus is to find the critical clearing time in the event of a disturbance. Direct methods, in particular, offer a way of evaluating the critical clearing time without explicitly solving the differential equations that model the electrical system by using energy functions.

Direct methods are an essential subject to understand this work contribution, as we develop a methodology to study uncertainties in power system using a direct method—called PEBS.

This chapter is an overview of direct methods for transient stability analysis, from its motivation—physical and mathematical—to a deduction of its essential formulae. Knowledgeable readers may skip this chapter or regard it as a quick reference.

4.1 Physical Motivational Example

The following example shows the motivation behind the use of direct methods. Imagine that a ball, initially resting at the bottom of a skate ramp, receives an impulse towards the edge. Depending on the magnitude of the impulse the ball will either fall—an unstable scenario—or rise to a certain height and then oscillate until it rests again at the bottom—a stable scenario. Given all problem conditions—mass, magnitude of impulse, etc., two methods can be used to compute how the disturbance will affect the system:

- Given the initial conditions, solve all differential equations that govern the ball dynamics and inspect in time domain how high it has risen;
- Evaluate how much energy the impulse has injected into the system and compare with the energy required to climb the ramp.

The second method is called a direct method because it assesses stability without explicitly solving the differential equations.

4.2 Mathematical Motivation

The last section example can be extended to a generic set of problems by using the Krasovskii-LaSalle invariance principle.

Theorem 4.2.1 (Krasovskii-LaSalle Invariance Principle) *Let $\Omega \subset D \subset \mathbb{R}^n$ be a compact positively invariant set with respect to (1.1). Let $V \rightarrow \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω . Let $E \subset \Omega$ be the set of all points in*

Ω where $\dot{V}(x) = 0$. Let $M \subset E$ be the greatest invariant set in E . Then, every solution starting in Ω approaches M as $t \rightarrow \infty$, i.e.,

$$\lim_{t \rightarrow \infty} \left(\inf_{z \in M} \underbrace{\|\phi(t, x_0) - z\|}_{\text{dist}(\phi(t, x_0), M)} \right) = 0.$$

Note that the inclusion of the sets in the theorem is $M \subset E \subset \Omega \subset D \subset \mathbb{R}^n$

A graphical representation of the theorem sets is shown in Figure 16.

Figure 16: Illustration of Krasovskii-LaSalle Invariance Principle

Theorem 4.2.1 introduces to us a methodology to find an estimate for the stability region. Let L be a constant, such that the level set $\Omega_L = \{x \in D : V(x) \leq L\}$ contains the equilibrium of (1.1) and it is contained in Ω . Then, by Theorem 4.2.1, every solutions starting in Ω_L approaches M as $t \rightarrow \infty$. In particular, if Ω_L is contained in the stability region of the equilibrium point, then Ω_L is an estimate for the stability region. Therefore, we can obtain a good estimate for the stability region by choosing a suitable constant L . In last section's, for example, L was chosen as the energy required to climb the ramp. We are now ready to introduce direct methods in power systems.

4.3 Direct Methods in Power Systems

To assess the transient stability of an electric power system, we observe the variation of rotor angles as a function of time. If the rotor angle of a machine or a group of machines starts to accelerate or decelerate with respect to the rest of the system, then the system is unstable. As developed before, the rotor angle of each machine is measured with respect to a fixed rotating frame, the synchronous reference. Therefore, instability indicates that the rotor angle of the machine, say the n -th machine, pulls away from the rest of the system. Hence, in order to solve this inconsistency, we need to watch relative rotor angles rather

than absolute rotor angles.

A power system experiencing a disturbance can be modeled by a set of three differential equations:

$$\dot{x}_{prf}(t) = f^{prf}(x_{prf}(t)) \quad t \in (-\infty, 0], \quad (4.1)$$

$$\dot{x}_f(t) = f^f(x_f(t)) \quad t \in (0, t_{cl}], \quad (4.2)$$

$$\dot{x}_{pf}(t) = f^{pf}(x_{pf}(t)) \quad t \in (t_{cl}, \infty), \quad (4.3)$$

where x_{prf} , x_f , x_{pf} and f^{prf} , f^f , f^{pf} are, respectively, the state variables and the dynamical equations of the system at pre-fault; fault-on and post-fault periods. This occurs because at $t = 0$, pre-fault period, the system experiences a fault and the dynamics change from the one driven by f^{prf} to f^f . For $t \in (0, t_{cl}]$, the fault-on period, the system dynamics is ruled by f^f . When protection acts and the fault is cleared, the post-fault period initiates. During this period the system is governed by f^{pf} .

Because in the pre-fault period the system is at equilibrium, $x_f(0) = x_{prf}$ ^I, and the initial condition for the post-fault system is $x_f(t_{cl})$, we have just:

$$\begin{aligned} \dot{x}_f(t) &= f^f(x_f(t)) \quad t \in (0, t_{cl}], \\ x_f(0) &= x_{prf}, \end{aligned} \quad (4.4)$$

and

$$\begin{aligned} \dot{x}_{pf}(t) &= f^{pf}(x_{pf}(t)) \quad t \in (t_{cl}, \infty), \\ x_{pf}(t_{cl}) &= x_f(t_{cl}), \end{aligned} \quad (4.5)$$

as the dynamical equations for the system. We can try to see this problem from a different perspective. Let x_e be an equilibrium of the post-fault system (4.5). Now the question is: will the trajectory $x_{pf}(t)$, with initial condition $x_f(t_{cl})$, converge to x_e as t goes to infinity? The largest value of t_{cl} for which this remains true is denominated the critical clearing time t_{cr} .

Consequently, if we have a good estimate of the stability region of the post-fault stable equilibrium point (sep) x_e , then t_{cr} is computed when the trajectory of (4.5) crosses the stability region boundary at x^* . Evaluating the stability region is, however, a difficult process, which requires several simulations of a complex system. Luckily, in the case of a power system, Corollary 1.4.1.1 shows that the stability region boundary is composed of the stable manifolds of the unstable equilibrium points (uep) of (4.4). To estimate the interior of the stability region of the post-fault system (4.4), we can use an inequality $V(x) < V_{cr}$, where $V(x)$ is an energy function for (4.4) and V_{cr} is a critical energy, distinct for each fault. Thus, instead of trying to find the stability region, one can find $V(x)$ and V_{cr} to estimate t_{cr} .

In the following section, we discuss a method that tries to compute $V(x)$ and V_{cr} called PEBS (potential energy boundary surface) method [2], [28].

^I Obtained by solving power-flow equations.

4.4 PEBS

In the PEBS method, the maximum of the potential energy parcel of $V(x)$ is used to calculate V_{cr} . For that purpose, we need to develop an expression for $V(x)$.

The swing equation for the i -th machine is as follows:

$$\frac{2H_i}{\omega_s} \ddot{\delta}_i = P_{m_i} - P_{e_i} - D_i \dot{\delta}_i, \quad i = 1, \dots, n, \quad (4.6)$$

where

$$P_{e_i} = E_i^2 G_{ii} + \sum_{\substack{j=1 \\ j \neq i}}^n [C_{ij} \sin \delta_{ij} + D_{ij} \cos \delta_{ij}]. \quad (4.7)$$

Defining $M_i := \frac{2H_i}{\omega_s}$ and $P_i := P_{m_i} - E_i^2 G_{ii}$, we obtain:

$$M_i \ddot{\delta}_i = P_i - D_i \dot{\delta}_i - \sum_{\substack{j=1 \\ j \neq i}}^n [C_{ij} \sin \delta_{ij} + D_{ij} \cos \delta_{ij}] \quad (4.8)$$

Furthermore, equation (4.8) can be compactly written as

$$M_i \ddot{\delta}_i + D_i \dot{\delta}_i = P_i - P_{e_i}(\delta_1, \dots, \delta_n), \quad i = 1, \dots, n. \quad (4.9)$$

Let us assume that θ_i is the rotor angle of the i -th machine with respect to a fixed reference. Then, $\theta_i = \delta_i + \omega_s t$; $\dot{\theta}_i = \omega_s + \dot{\delta}_i$ and $\dot{\delta}_i = \dot{\theta}_i - \omega_s$, where ω_s is the synchronous speed in radians per sec. Thus, both δ and $\dot{\delta}$ are referenced to a synchronously rotating frame. Denoting $\hat{\omega}_i := \dot{\delta}_i$, equation (4.9) is transformed into a set of first-order differential equations:

$$\dot{\delta}_i = \hat{\omega}_i - \omega_s \quad (4.10)$$

$$\dot{\omega}_i = \frac{1}{M_i} [P_i - P_{e_i}(\delta_1, \dots, \delta_n) - D_i(\hat{\omega}_i - \omega_s)], \quad i = 1, \dots, n \quad (4.11)$$

Moreover, we can simplify (4.10) and (4.11) by defining $\omega_i := \hat{\omega}_i - \omega_s$. Hence, ω_i is now the difference between angular velocity of the rotor and synchronous speed.

To apply equations (4.10) and (4.11) to both fault-on and post-fault periods, we only need to change P_{e_i} . This happens because the admittance matrix is different for each period.

We formulate the energy function using the center of inertia (COI)^{II} as reference. The COI represents a mean motion of the system in a similar fashion as the center of mass represents an average position of all the parts of a body. The formulation for the center of inertia is as follows:

$$\delta_o = \frac{1}{M_T} \sum_{i=1}^n M_i \delta_i \text{ and } \omega_o = \frac{1}{M_T} \sum_{i=1}^n M_i \omega_i, \quad (4.12)$$

^{II} Also known as center of angle (COA).

where $M_T = \sum_{i=1}^n M_i$.

Differentiating equation (4.12), we find

$$\dot{\omega}_o = \frac{1}{M_T} \sum_{i=1}^n P_{m_i} - P_{e_i}. \quad (4.13)$$

Substituting (4.7) in (4.13), we obtain

$$\sum_{i=1}^n \left[P_{m_i} - E_i^2 G_{ii} + \sum_{\substack{j=1 \\ \neq i}}^n [C_{ij} \sin \delta_{ij} + D_{ij} \cos \delta_{ij}] \right]. \quad (4.14)$$

Because summation is a linear operator, we can write

$$\sum_{i=1}^n [P_{m_i} - E_i^2 G_{ii}] + \sum_{i=1}^n \sum_{\substack{j=1 \\ \neq i}}^n [C_{ij} \sin \delta_{ij} + D_{ij} \cos \delta_{ij}]. \quad (4.15)$$

Opening the second term of (4.15), we obtain two parts (I and II) as follows:

$$\underbrace{\sum_{i=1}^n \sum_{\substack{j=1 \\ \neq i}}^n C_{ij} \sin \delta_{ij}}_I + \underbrace{\sum_{i=1}^n \sum_{\substack{j=1 \\ \neq i}}^n D_{ij} \cos \delta_{ij}}_{II} \quad (4.16)$$

Expanding the first summation in part I, we get

$$\sum_{\substack{j=1 \\ \neq 1}}^n C_{1j} \sin \delta_{1j} + \sum_{\substack{j=1 \\ \neq 2}}^n C_{2j} \sin \delta_{2j} + \cdots + \sum_{\substack{j=1 \\ \neq n}}^n C_{nj} \sin \delta_{nj}. \quad (4.17)$$

Expanding (4.17), we have the following expression:

$$(C_{12} \sin \delta_{12} + C_{13} \sin \delta_{13} + \cdots + C_{1n} \sin \delta_{1n}) + (C_{21} \sin \delta_{21} + C_{23} \sin \delta_{23} + \cdots + C_{2n} \sin \delta_{2n}) + \cdots + (C_{n1} \sin \delta_{n1} + C_{n2} \sin \delta_{n2} + \cdots + C_{nn-1} \sin \delta_{nn-1}) \quad (4.18)$$

By virtue of (2.30), (2.31) and (2.64), we know that $C_{ij} = C_{ji}$. Thus, we can rewrite (4.18) as

$$C_{12}(\sin \delta_{12} + \sin \delta_{21}) + C_{13}(\sin \delta_{13} + \sin \delta_{31}) + \cdots + C_{nn-1}(\sin \delta_{nn-1} + \sin \delta_{n-1n}). \quad (4.19)$$

Because sine is odd, the sum is zero. Developing in the same manner for the second part and knowing that cosine is even, we find that the sum II is:

$$2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n D_{ij} \cos \delta_{ij} \quad (4.20)$$

Finally, the expression for P_{COI} is as follows:

$$P_{COI} = \sum_{i=1}^n [P_{m_i} - E_i^2 G_{ii}] - 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n D_{ij} \cos \delta_{ij} \quad (4.21)$$

The state variables with respect to the new reference are $\theta_i = \delta_i - \delta_o$ and $\tilde{\omega}_i = \omega_i - \omega_o$. Differentiating θ_i , we have:

$$\dot{\theta}_i = \dot{\theta}_i - \dot{\delta}_o \quad (4.22)$$

$$= \omega_i - \omega_o \quad (4.23)$$

$$= \tilde{\omega}_i \quad (4.24)$$

Assuming that the system has no damping, i.e., $D_i = 0$, the swing equation (4.9) becomes:

$$M_i \ddot{\theta}_i = P_i - P_{e_i}(\theta_i, \dots, \theta_n) - \frac{M_i}{M_T} P_{COI} := f_i(\theta), \quad i = 1, \dots, n. \quad (4.25)$$

Let the post-fault equilibrium point be $[\theta, \tilde{\omega}_i]^T = [\theta^e, 0]^T$. We can compute θ^e by solving

$$f_i(\theta) = 0, \quad i = 1, \dots, n. \quad (4.26)$$

Due to the definition of θ_i , we have that $\sum_{i=1}^n M_i \theta_i = 0$. Therefore, the differential equations are linearly dependent and θ_n can be written as function of θ_i , $i = 1, \dots, n$. Lastly, we can rewrite equation (4.26) as follows:

$$f_i(\theta_1, \theta_2, \dots, \theta_{n-1}) = 0, \quad i = 1, \dots, n-1. \quad (4.27)$$

To find an energy function we shall proceed as in [29].

First we write the differential equations (4.26) into the symmetric form:

$$dt = \frac{M_1 d\tilde{\omega}_1}{f_1(\theta)} = \frac{M_2 d\tilde{\omega}_2}{f_2(\theta)} = \dots = \frac{M_n d\tilde{\omega}_n}{f_n(\theta)}. \quad (4.28)$$

Noting that $\frac{M_1 d\tilde{\omega}_i}{f_i(\theta)} = \frac{d\theta_i}{\tilde{\omega}_i}$, $i = 1, \dots, n$, we have:

$$dt = \frac{M_1 d\tilde{\omega}_1}{f_1(\theta)} = \frac{d\theta_1}{\tilde{\omega}_1} = \frac{M_2 d\tilde{\omega}_2}{f_2(\theta)} = \frac{d\theta_2}{\tilde{\omega}_2} = \dots = \frac{M_n d\tilde{\omega}_n}{f_n(\theta)} = \frac{d\theta_n}{\tilde{\omega}_n} \quad (4.29)$$

We can check the consistency of equation (4.29) by using dimensional analysis:

$$s = \frac{\text{kg} \cdot \text{m}^2 \cdot \text{rad} \cdot \text{s}^{-1}}{\text{kg} \cdot \text{m}^2 \cdot \text{rad} \cdot \text{s}^{-2}} = \frac{\text{rad}}{\text{rad} \cdot \text{s}^{-1}} \rightarrow s = s, \quad (4.30)$$

so it is dimensionally consistent.

Integrating the pairs of equations for each machine between the post-fault sep, $[\theta_i^e, 0]^T$, to $[\theta_i, 0]^T$ yields

$$V_i(\theta, \tilde{\omega}_i) = \frac{1}{2} M_i \tilde{\omega}_i^2 - \int_{\theta_i^e}^{\theta_i} f_i(\theta) d\theta_i, \quad i = 1, \dots, n. \quad (4.31)$$

Equation (4.31) is the energy of a single machine. Summing the energy for all machines, we obtain the energy of the system:

$$V(\theta, \tilde{\omega}) = \frac{1}{2} \sum_{i=1}^n M_i \tilde{\omega}_i^2 - \sum_{i=1}^n P_i(\theta_i - \theta_i^e) \quad (4.32)$$

$$- \sum_{i=1}^{n-1} \sum_{j=1+1}^n \left[C_{ij} (\cos \theta_{ij} - \cos \theta_{ij}^e) - \int_{\theta_i^e + \theta_j^e}^{\theta_i + \theta_j} D_{ij} \cos \theta_{ij} d(\theta_i + \theta_j) \right] \\ = V_k(\tilde{\omega}) + V_p(\theta), \quad (4.33)$$

since

$$\sum_{i=1}^n \frac{M_i}{M_T} \int_{\theta_i^e}^{\theta_i^u} \frac{M_i}{M_T} P_{COI} d\theta_i = \int_0^t \sum_{i=1}^n P_{COI} \left(\frac{d\theta_i}{dt} \right) dt = \sum_{i=1}^n \frac{M_i}{M_T} P_{COI} \tilde{\omega}_i dt = 0. \quad (4.34)$$

Looking at equation (4.32), we can verify the existence of path-dependent integral terms. Thus, the system is not conservative and we cannot argue that V is positive definite. However, if $D_{ij} = 0$, one can show that $V(\theta, \tilde{\omega})$ is an energy function [28], [30], [31].

Assuming that we are calculating the energy at $[\theta^u, \tilde{\omega}]^T$, an approximation for the path-dependent integral is given in [32], [33]:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n D_{ij} \frac{\theta_i^u + \theta_j^u - \theta_i^e - \theta_j^e}{\theta_i^u - \theta_j^u - \theta_i^e + \theta_j^e} \left[\sin(\theta_i^u - \theta_j^u) - \sin(\theta_i^e - \theta_j^e) \right]. \quad (4.35)$$

To derive expression (4.35), we proceed as follows. Consider a line segment from θ_i^e to θ_i^u , then any point on the segment is $\theta_i = \theta_i^e + p(\theta_i^u - \theta_i^e)$, $p \in [0, 1]$. Hence, $d(\theta_i + \theta_j) = dp(\theta_i^u - \theta_i^e + \theta_j^u - \theta_j^e)$, and the integral becomes:

$$\left(\theta_i^u - \theta_i^e + \theta_j^u - \theta_j^e \right) \int_0^1 D_{ij} \cos \left[\left(\theta_i^e - \theta_j^e \right) + p \left(\left(\theta_i^u - \theta_i^e \right) - \left(\theta_j^u - \theta_j^e \right) \right) \right] dp. \quad (4.36)$$

Using that $\int_{x_0}^{x_1} \cos(a + bx) dx = \left[\frac{\sin(a+bx)}{b} \right]_{x=x_0}^{x=x_1}$, we obtain:

$$\frac{(\theta_i^u - \theta_i^e) + (\theta_j^u - \theta_j^e)}{(\theta_i^u - \theta_i^e) - (\theta_j^u - \theta_j^e)} D_{ij} \left[\sin \left(\left(\theta_i^e - \theta_j^e \right) + p \left(\left(\theta_i^u - \theta_i^e \right) - \left(\theta_j^u - \theta_j^e \right) \right) \right) \right]_{p=0}^{p=1}. \quad (4.37)$$

Finally, by evaluating (4.37), we arrive at the final expression:

$$\frac{(\theta_i^u - \theta_i^e) + (\theta_j^u - \theta_j^e)}{(\theta_i^u - \theta_i^e) - (\theta_j^u - \theta_j^e)} D_{ij} \left[\sin(\theta_i^u - \theta_j^u) - \sin(\theta_i^e - \theta_j^e) \right]. \quad (4.38)$$

4.4.1 Calculating V_{cr}

First, let us consider the simplest case of a machine connected to an infinite bus. As the inertia of an infinite bus is infinite, we can say that $\delta_o = \delta_\infty$, $\omega_o = 0$ ^{III} and $\frac{M_i}{M_T} P_{COI} = 0$ if $i \neq \infty$ and $\frac{M_i}{M_T} P_{COI} = 1$ if $i = \infty$. Therefore, $\tilde{\omega}_1 = \omega_1$, $\theta_1 = \delta_1 - \delta_\infty$, $\tilde{\omega}_\infty = 0$, $\theta_\infty = 0$. Furthermore, we obtain the original swing equation by choosing $\delta_\infty = 0$:

$$\dot{\delta} = \omega \quad (4.39)$$

$$\dot{\omega} = \frac{1}{M} \left[P_m - E^2 G + EE_\infty G \cos \delta + EE_\infty \sin \delta \right] \quad (4.40)$$

Assuming that $[\delta^e, 0]^T$ is the post-fault equilibrium, we obtain using (4.33):

$$\begin{aligned} V(\delta, \omega) = \frac{1}{2} M \omega^2 - P_m(\delta - \delta^e) + E^2 G(\delta - \delta^e) + EE_\infty B(\cos \delta - \cos \delta^e) \\ - EE_\infty G(\sin \delta - \sin \delta^e) = V_k(\omega) + V_p(\delta). \end{aligned} \quad (4.41)$$

^{III} Remark: There is no deviation in frequency.

To further study equation (4.41), let us assume that there are no resistive loss on the transmission line and the potential parcel of the energy function can be written as follows:

$$V_p(\delta) = -P_m(\delta - \delta^e) + EE_\infty B(\cos \delta - \cos \delta^e) = -P_m(\delta - \delta^e) - P_{\max}(\cos \delta - \cos \delta^e) \quad (4.42)$$

Figure 17: PEBS

Plotting equation (4.42), Figure 17, we see that around the stable equilibrium, $V_p(\delta^e) = 0$, there is a potential well bounded by two relative maximums^{IV}. This potential boundary is called PEBS and it is similar to the motivational example first given. Thus, assuming that the fault is cleared at $[\delta_{cl}, \omega_{cl}]^T$, to determine if the post-fault system is stable, we need to verify if it can absorb the excesses of kinetic energy, i.e.,

$$V_p(\delta^u) - V_p(\delta_{cl}) > V_k(\omega_{cl}). \quad (4.43)$$

The critical case will occur when the inequality in (4.43) is substituted by an equality. In this case, we have:

$$V_p(\delta^u) = V_{cr} = V_k(\omega_{cl}) + V_p(\delta_{cl}), \quad (4.44)$$

where V_{cr} is the critical energy and $[\delta_{cl}, \omega_{cl}]^T$ is the critical clearing point. Hence, we calculate V_{cr} at the most relevant unstable equilibrium point for the system. Literature refers to this point as being the controlling uep (δ^{cuep}) [32], [34].

Extending these concepts to a multi-machine system is not a trivial task. Now the PEBS bounds a multidimensional “well” formed by the unstable equilibriums of the system and by lines, orthogonal to the equipotential curves, connecting them [28]. Moreover, the point

^{IV} In fact, the relative maximums of potential are at the unstable equilibriums points, namely $\delta^u, \hat{\delta}^u = \pm\pi - \delta^e$.

where the faulted trajectory crosses the PEBS now is called exit-point, θ^* and it is not the controlling uep (θ^{cuep}). In reality, we consider that $V_p(\theta^*) \approx V_p(\theta^{cuep})$, which will not always be the case. The algorithm for executing the PEBS method is as follows [29]:

- Integrate the fault-on trajectory until V_p reaches a maximum, $V_p^{\max}$ in time. Let $V_{cr} = V_p^{\max}$;
- From the fault-on trajectory find the time at which $V(\theta, \tilde{\omega}) = V_{cr}$. This time is an estimate for t_{cr} .

In the following chapter, we will present a version of the PEBS method that includes uncertainties in the analysis. Henceforth, we shall call this version as robust PEBS.

5 METHODOLOGIES

In this chapter, we will discuss the methodologies developed in this work to assess the transient stability of a power system with uncertainties.

We begin by commenting that the inclusion of uncertainties in the transient stability problem is not a trivial task. A myriad of papers tried to tackle this problem differently. Chiodo, Gagliardi, and Lauria [35], for example, have developed an analytical expression for the lognormal distribution of the critical clearing time for a single machine infinite bus system with a normally distributed load. Timko, Bose, and Anderson [36], on the other hand, have applied the Monte Carlo method to assess the critical clearing time probability distribution.

A major issue in several of the aforementioned works is that they force us to choose too harshly between precision and computational time. A brute force search by a Monte Carlo simulation will surely give a precise result with a high computational cost, while approximating a multi-machine system to a single machine one, as proposed in [37] and [38], will drastically decrease the computation time but it will reduce the result precision as well.

In this work, we deal with this apparent irreconcilability using three different strategies:

- proposing an extension of the PEBS method called robust PEBS;
- using optimization techniques;
- choosing the most influential parameters using sensitivity analysis.

In the first strategy, we assume that all uncertainties are modeled as interval and extend the PEBS method to include them in a natural manner. A considerable advantage of this method, called robust PEBS, is that the use intervals allowed us to find closed expressions for the stability analysis. Neither iterations nor simulations are needed to find the critical clearing time. Another advantage, inherently from direct methods, is that in the robust PEBS we only need to integrate the fault-on period—the shortest period in the analysis—to assess the power system stability.

The robust PEBS can, however, lead to over conservative assessments. If a precise value for the critical clearing time is needed, then it may not perform as desired. Our second strategy deals with this problem by employing optimization techniques, namely Simulated Annealing and Differential Evolution, to find an accurate value for the critical clearing time. This strategy greatly outperforms the classical Monte Carlo, since few iterations are wasted in regions where the minimum critical clearing time is unlikely to be found.

The third strategy is actually a complementary technique: it tries to find the most

influential parameters for the transient stability assessment using a sensitivity analysis. This parameter screening reduces the number of uncertainties, decreasing the search space for the optimization routines and cutting down the number of intervals in the robust PEBS expressions. As shall be seen in the Results chapter, the proposed methodology greatly accelerates the simulation with an unexpressive impact on the results.

This chapter is divided as the following: in section 5.1, we introduce the robust PEBS method; in section 5.2, we present the optimization routines and discuss their application to the transient stability analysis problem; finally, in section 5.3, we present a methodology to find the most influential parameters for the transient analysis assessment using the sensitivity analysis.

5.1 Robust PEBS

In this section, we introduce a robust version of the PEBS method—firstly presented in [3]—and subsequently propose an interval extension for the energy function. In the first part of the section, we present the required mathematical motivation for developing the method. In the second part, we develop the robust PEBS method for a single machine connected to an infinite bus with only post-fault uncertainties, and in sequence we generalize the method for a multi-machine system with uncertainties in all periods. Lastly, we discuss the method implementation details.

5.1.1 Mathematical Background

While the PEBS method is based on the Krasovskii-LaSalle invariance principle (Theorem 4.2.1), the robust version is based on a Uniform Version of the Invariance Principle.

For the Uniform Invariance Principle, let us consider the following autonomous system:

$$\dot{x} = f(x, \lambda), \quad (5.1)$$

where $\lambda \in \Lambda \subset \mathbb{R}^m$ is a parameter vector of this system and $x \in \mathbb{R}^n$ is the state vector. The set Λ is constructed as being $\Lambda = [\lambda_1^{\min}, \lambda_1^{\max}]^T \times [\lambda_2^{\max}, \lambda_2^{\min}]^T \times \dots \times [\lambda_m^{\min}, \lambda_m^{\max}]^T$, in which $\lambda_1^{\min}, \lambda_2^{\min}, \dots, \lambda_m^{\min}$ and $\lambda_1^{\max}, \lambda_2^{\max}, \dots, \lambda_m^{\max}$ represent, respectively, minimum and maximum of each parameter $\lambda_1, \lambda_2, \dots, \lambda_m$ and therefore Λ is a box.

The principle is stated as follows:

Theorem 5.1.1 (Uniform Invariance Principle [3]) *Suppose $f : \mathbb{R}^n \times \Lambda \rightarrow \mathbb{R}^n$ and $V : \mathbb{R}^n \times \Lambda \rightarrow \mathbb{R}$ are continuously differentiable functions and $a, b, c : \mathbb{R}^n \rightarrow \mathbb{R}$ are continuous functions. Assume that for any $[x, \lambda]^T \in \mathbb{R}^n \times \Lambda$, one has:*

$$a(x) \leq V(x, \lambda) \leq b(x), \quad -\dot{V}(x, \lambda) \geq c(x). \quad (5.2)$$

For $L > 0$, let $A_L := \{x \in \mathbb{R}^n : a(x) < L\}$. Assume that A_L is nonempty and bounded and consider the sets:

$$B_L := \{x \in \mathbb{R}^n : b(x) < L\}, C := \{x \in \mathbb{R}^n : c(x) < 0\} \quad \text{and } E_L := \{x \in A_L : c(x) = 0\}. \quad (5.3)$$

Suppose now that $\sup_{x \in C} b(x) \leq l < L$ and define the sets:

$$A_l := \{x \in \mathbb{R}^n : a(x) \leq l\} \text{ and } B_l := \{x \in \mathbb{R}^n : b(x) \leq l\}. \quad (5.4)$$

If λ is a fixed parameter in Λ and all the previous conditions are satisfied, then for $x_0 \in B_L$ the solution $\phi(t, x_0, \lambda)$ is defined in $[0, \infty)$ and the following holds:

- I. if $x_0 \in B_l$ then $\phi(t, x_0, \lambda) \in A_l$, for $t > 0$ and $\phi(t, x_0, \lambda)$ tends to the largest invariant set of (5.1) contained in A_l , as $t \rightarrow \infty$;
- II. if $x_0 \in B_L \setminus B_l$ then $\phi(t, x_0, \lambda)$ tends to the largest invariant set of (5.1) contained in $A_l \cup E_L$.

Figure 18 illustrates the sets from Theorem 5.1.1 and Figure 19 shows a graphical representation of the energy functions $a(x)$, $b(x)$ and $c(x)$.

Figure 18: Uniform Invariance Principle Sets

In the same manner that Theorem 4.2.1 motivated us to use level sets of an energy function V to estimate the stability region, Theorem 5.1.1 motivate us to find the functions a , b to bound V and find a robust estimate for the stability region. Finding such function is not a trivial task, nevertheless we will show that by using interval arithmetics their appear naturally.

A proof and further details of this theorem can be found in [39].

Figure 19: Uniform Invariance Principle — Functions $a(x)$, $b(x)$ and $c(x)$

Source: Adapted from Bretas and Alberto [3]

5.1.2 Robust PEBS for a Single Machine Infinity Bus System

Starting with the simplest case of a single machine infinite bus power system with uncertainties in the post-fault period, Theorem 5.1.1 will be used for the stability. Let us consider that $\lambda = [M, P_m, E, E_\infty, G, B]$ is the post-fault parameter vector and L is V_{cr} . Thus, we rewrite equation (4.41) as being:

$$V(\delta, \omega, \lambda) = \frac{M}{2}\omega^2 - P_m + E^2 G \delta + E E_\infty B \cos(\delta) - E E_\infty G \sin(\delta) + k, \quad (5.5)$$

where k is an arbitrary constant and

$$V_k(\omega, \lambda) = \frac{M}{2}\omega^2, \quad (5.6)$$

$$V_p(\delta, \lambda) = -P_m \delta + E^2 G \delta + E E_\infty B \cos(\delta) - E E_\infty G \sin(\delta) + k. \quad (5.7)$$

The derivative of V is given by^I:

$$\dot{V}(\delta, \omega, \lambda) = -D^2 \omega. \quad (5.8)$$

The key point now is how to find $a(\delta, \omega)$, $b(\delta, \omega)$ and $c(\delta, \omega)$. We propose the use of an interval extension of V and $\dot{V}$ to find such functions.

5.1.2.1 Interval Extension of the Energy Function

Consider that the post-fault system has parametric uncertainties represented by intervals. The energy function for the post-fault system evaluated at the fault-on trajectory is then dependent of the parameters, i.e.,

$$V(x, \Lambda), \quad (5.9)$$

^I Considering that the system has damping.

where $x \in \mathbb{R}^n$ is the state vector of the faulted system and $\Lambda \in \mathbb{I}^m$ is the interval parameter vector. Thus, $V : \mathbb{R}^n \times \mathbb{I}^m \rightarrow \mathbb{I}$ is an interval function and its natural interval extension is:

$$\mathcal{V}(x, \Lambda) = [\underline{\mathcal{V}}(x, \Lambda), \overline{\mathcal{V}}(x, \Lambda)]. \quad (5.10)$$

Using the Fundamental Theorem of Interval Analysis (Theorem 3.3.1), we can write the following expression:

$$\underline{\mathcal{V}}(x, \Lambda) \leq V(x, \Lambda) \leq \overline{\mathcal{V}}(x, \Lambda). \quad (5.11)$$

Clearly, it is possible to use $\underline{\mathcal{V}}(x, \Lambda)$ as a and $\overline{\mathcal{V}}(x, \Lambda)$ as b in Theorem 5.1.1. Hence, interval arithmetic naturally finds the bounds of V .

In particular, for the one machine infinite bus system we have:

$$\mathcal{V}(\delta, \omega, \Lambda) = \frac{\Lambda_1}{2} \cdot \omega^2 - \Lambda_2 \cdot \delta + \Lambda_3^2 \cdot \Lambda_5 \cdot \delta + \Lambda_3 \cdot \Lambda_4 \cdot \Lambda_6 \cdot \cos(\delta) - \Lambda_4 \cdot \Lambda_1 \cdot \sin(\delta), \quad (5.12)$$

where $\Lambda = [\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4, \Lambda_5, \Lambda_6]^\top$, $\Lambda_1 = [\underline{M}, \overline{M}]$, $\Lambda_2 = [\underline{P_m}, \overline{P_m}]$, $\Lambda_3 = [\underline{E}, \overline{E}]$, $\Lambda_4 = [\underline{E_\infty}, \overline{E_\infty}]$, $\Lambda_5 = [\underline{G}, \overline{G}]$ and $\Lambda_6 = [\underline{B}, \overline{B}]$.

To show that (5.12) is an interval extension of (5.5), just choose $\Lambda_i = [\lambda_i, \lambda_i]$, $\lambda_i \in \mathbb{R}$ for $i = 1, 2, \dots, 6$ and apply to $\mathcal{V}$, wherewith we obtain:

$$\begin{aligned} \mathcal{V}(\delta, \omega, \Lambda) = & \frac{[\lambda_1, \lambda_1]}{2} \cdot \omega^2 - [\lambda_2, \lambda_2] \cdot \delta + [\lambda_3, \lambda_3]^2 \cdot [\lambda_5, \lambda_5] \cdot \delta + [\lambda_3, \lambda_3] \cdot [\lambda_4, \lambda_4] \cdot [\lambda_6, \lambda_6] \cdot \cos(\delta) \\ & - [\lambda_4, \lambda_4] \cdot [\lambda_1, \lambda_1] \cdot \sin(\delta). \end{aligned} \quad (5.13)$$

Using the interval operations $\langle \mathbb{I}, +, -, \cdot, / \rangle$ as defined in chapter 3, we can write:

$$\begin{aligned} \mathcal{V}(\delta, \omega, \Lambda) = & \left[\frac{\lambda_1 \omega^2}{2} - \lambda_2 \delta + \lambda_3^2 \lambda_5 \delta + \lambda_3 \lambda_4 \lambda_6 \cos(\delta) - \lambda_4 \lambda_1 \sin(\delta), \frac{\lambda_1 \omega^2}{2} - \lambda_2 \delta + \lambda_3^2 \lambda_5 \delta \right. \\ & \left. + \lambda_3 \lambda_4 \lambda_6 \cos(\delta) - \lambda_4 \lambda_1 \sin(\delta) \right]. \end{aligned} \quad (5.14)$$

Noting that (5.14) is a degenerate interval, we get:

$$\mathcal{V}(\delta, \omega, \Lambda) = \frac{\lambda_1 \omega^2}{2} - \lambda_2 \delta + \lambda_3^2 \lambda_5 \delta + \lambda_3 \lambda_4 \lambda_6 \cos(\delta) - \lambda_4 \lambda_1 \sin(\delta). \quad (5.15)$$

Thus,

$$\mathcal{V}(\delta, \omega, \Lambda) = V(\delta, \omega, [\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6]), \quad (5.16)$$

and (5.12) is an interval extension of (5.5). In addition, since $\mathcal{V}$ was obtained by replacing all real operations by their interval equivalents, equation (5.12) is a natural interval extension of (5.5).

Since $D > 0$:

$$-\overline{D}^2 \omega \leq \dot{V}(\delta, \omega, \Lambda) \leq -\underline{D}^2 \omega. \quad (5.17)$$

Inequation 5.17 shows that we can choose c to be $\overline{D}^2 \omega$.

5.1.2.2 Algorithm

Since c is not negative, sets C , A_l , and B_l are empty. Then, the conclusion II of Theorem 5.1.1 can be used to guarantee the stability of the post-fault equilibrium point and the set B_L will be a stability region estimate. Similarly to the original PEBS method, we have chosen V_{cr} to be the first maximum of a_p (potential parcel of a). But, instead of using a to find the critical clearing time, we use^{II} b . Hence, the method can be summarized in the following steps:

- Integrate the fault-on trajectory until a_p reaches a maximum, $a_p^{\max}$ in time. Let $V_{cr} = a_p^{\max}$;
- From the fault-on trajectory find when $b = V_{cr}$. This is an estimate for t_{cr} .

This is the simplest case, not only because we are dealing with only a single machine, but also because we do not need to integrate the post-fault uncertain system—as the PEBS method only requires integrating the fault-on system.

5.1.3 Robust PEBS for a Multi-Machine Power System

In this section, we generalize the concepts presented in the previous section to a multi-machine system considering uncertainties at all periods and subsequently discuss the implementation details of the method.

We begin by devising which parameters have uncertainty. The machines model/parameters are well known, so it is safe to assume that they are not uncertain. On the other hand, the line parameters, resistance and reactance, may have an error. The loads may also have uncertainty; the reason for this is: the loads may represent a system with distributed generation and since this type of generation have low inertia or are controlled by power electronics—have a faster dynamic than the transient stability problem—one can model them as a static generation with an uncertainty.

Following the same steps as before, consider that the power system has parametric uncertainties that can be represented by intervals. The uncertain system dynamic can be described as:

$$\begin{aligned}
 \dot{X}_f(t) &= f^f(X_f(t), \Lambda) \quad t \in (0, t_{cl}], \\
 X_f(0) &= X_{prf}, \\
 \dot{X}_{pf}(t) &= f^{pf}(X_{pf}(t), \Lambda) \quad t \in (t_{cl}, \infty), \\
 X_{pf}(t_{cl}) &= X_f(t_{cl}),
 \end{aligned} \tag{5.18}$$

where $X_f, X_{pf} \in \mathbb{I}^n$ and $f^f, f^{pf} : \mathbb{I}^n \times \mathbb{I}^m \rightarrow \mathbb{I}^n$ are, respectively, the interval state vector and the interval differential equations of the system at fault-on and post-fault periods;

^{II} This happens because of the definition of B_L as stated in Theorem 5.1.1.

$\Lambda \in \mathbb{I}^m$ is the interval parameter vector; and $X_{prf} \in \mathbb{I}^n$ is the pre-fault interval state vector. Since the power system now has uncertainties at all periods, not only the fault-on and post-fault ordinary differential equations became interval differential equations, but also the power flow solution for the pre-fault period is an interval vector. Thus, differently from the former section, now both arguments of the energy function evaluated at the fault-on trajectory are intervals. Assuming, that $X_f = [\Theta, \Omega]^\top$, the previous statement can be mathematically written as:

$$V(\Theta, \Omega, \Lambda). \quad (5.19)$$

Thus, $V : \mathbb{R}^n \times \mathbb{I}^m \rightarrow \mathbb{I}$ is an interval function and its interval extension is:

$$\mathcal{V}(\Theta, \tilde{\Omega}, \Lambda) = \frac{1}{2} \sum_{i=1}^n M_i \cdot \tilde{\Omega}_i^2 - \sum_{i=1}^n [\underline{P}_i, \overline{P}_i] \cdot (\Theta_i - \Theta_i^e) \quad (5.20)$$

$$- \sum_{i=1}^{n-1} \sum_{j=1+1}^n \left[[\underline{C}_{ij}, \overline{C}_{ij}] \cdot (\cos \Theta_{ij} - \cos \Theta_{ij}^e) \right] \quad (5.21)$$

$$- [\underline{D}_{ij}, \overline{D}_{ij}] \cdot \frac{\Theta_i + \Theta_j - \Theta_i^e - \Theta_j^e}{\Theta_i - \Theta_j - \Theta_i^e + \Theta_j^e} \cdot (\sin \Theta_{ij} - \sin \Theta_{ij}^e) \right]. \quad (5.22)$$

Using the Fundamental Theorem of Interval Analysis (Theorem 3.3.1), we can write the following expression:

$$\underline{\mathcal{V}}(\Theta, \Omega, \Lambda) \leq V(\Theta, \Omega, \Lambda) \leq \overline{\mathcal{V}}(\Theta, \Omega, \Lambda). \quad (5.23)$$

Once more, it is possible to choose $\underline{\mathcal{V}}(\Theta, \Omega, \Lambda)$ as a and $\overline{\mathcal{V}}(\Theta, \Omega, \Lambda)$ as b for Theorem 5.1.1. Since the expression of (5.22) was obtained by changing every real variable by an interval variable and every real operation by its equivalent interval operation, function (5.22) is a natural interval extension of (4.33).

The only problem in (5.22) is that we are unable to show that the denominator in the path dependent integral approximation does not contain zero. To overcome this challenge, two possible solutions exist:

1. Separate the interval that contains zero in two half-closed intervals and operate the division individually;
2. Separate the interval that contains zero in two half-closed intervals, operate the division individually, and take the interval hull of the results.

The second solution was preferred as it is slower to carry two half-closed intervals in all other operations than to use only one closed interval.

Finally, the steps for executing this method follow the same fashion as the former, and they can be summarized as bellow:

- Integrate the fault-on trajectory until a_p reaches a maximum, $a_p^{\max}$ in time. Let $V_{cr} = a_p^{\max}$;

- From the fault-on trajectory find the time at which $b = V_{cr}$. This time is an estimate for t_{cr} .

Figure 20 illustrates a graphical representation of the robust PEBS method.

Figure 20: Robust PEBS Method – Graphical Representation

5.1.3.1 Multi-Machine Robust PEBS Implementation

In this section, a discussion of the implementation details of the multi-machine robust PEBS method is carried out.

Two major numerical operations are needed for executing the robust PEBS method: Solve an interval dynamical equation and solve an interval nonlinear system. In Chapter 3, it was introduced the methods to perform both numerical operations, therefore we will now focus on how we implemented them.

In subsection 3.5.2, two methodologies were presented to solve an interval nonlinear system: Newton's and nonlinear Krawczyk's method. We have selected the former in an initial approach, as the nonlinear equations that need to be solved are the power flow equations, we just need to extend them and the well known Jacobian matrix to intervals equations and an interval matrix, respectively. The downside from using Newton's method is the

necessity of solving at each iteration a linear system of equations. Therefore, we have to use or the Krawczyk method or the Gauss-Seidel method—both discussed in section 3.5.1. The last was preferred since it produced the sharpest enclosure for the solution in our tests. Nonetheless, to add robustness to our code, we adopted the following execution sequence:

- I. Solve the nonlinear system using Newton’s method with Gauss-Seidel method and store the result;
- II. If the solution converged, return the result and end the program. Else, continue;
- III. Solve the nonlinear system using Newton’s method with Krawczyk method and store the result;
- IV. If the solution converged, return the result and end the program. Else, continue;
- V. Solve the nonlinear system using nonlinear Krawczyk’s method and store the result;
- VI. If the solution converged, return the result and end the program. Else, continue;
- VII. Return the sharpest solution of all three and end the program.

Finally, to solve an interval differential equation, we employed the Numerical Interval Simulation method as discussed in subsection 3.5.3.

5.2 Optimization Methods

In this section, we discuss two methods to find the critical clearing time of power systems with uncertainties using the classical method of integrating the fault-on and post-fault system in time. The aim when developing such methodologies is to provide balance between the robust PEBS execution speed and the Monte Carlo precision.

We remark that Monte Carlo methods are widely used in probabilistic transient stability analysis, [36], [37], [40], [41]. Usually, the inputs for the method are the events that can cause instability—a fault for example—and the power system parameters and the result is the probability distribution of the critical clearing time.

Unfortunately, despite generating accurate and complete results, this approach requires a vast amount of information about the system and an even greater number of simulations for its execution. Thus, it is suitable for generating reference results and not real-time applications.

Since we are not interested in the critical clearing time probability distribution, a Monte Carlo simulation wastes computational time by needlessly evaluating the critical clearing in regions of the parameter space where the minimum is non existent. It is clear, therefore, that a better approach is needed.

If we again consider that the power system parameters are intervals, then it is possible

to the problem of finding the minimum critical clearing time as a minimization problem. Differently, from the usual optimization problems, the function we are trying to minimize is not a mathematical function per se, but a computational one that evaluates the critical clearing time given a set of parameters. Mathematically, this statement can be written as:

$$\begin{aligned} & \underset{\lambda}{\text{minimize}} && f(\lambda) \\ & \text{subject to} && \lambda_i^{\min} \leq \lambda_i \leq \lambda_i^{\max}, \quad i = 1, 2, \dots, m. \end{aligned} \tag{5.24}$$

where $f : \mathbb{R}^m \rightarrow \mathbb{R}$ represents the computational function that evaluates the critical clearing time; $\lambda \in \Lambda \subset \mathbb{R}^m$ is the parameter vector; and $\lambda_i^{\min}$, $\lambda_i^{\max}$ are, respectively the maximum and minimum of the i -th parameter.

The function clearly does not have an analytical derivative, so to employ methods such as Broyden–Fletcher–Goldfarb–Shanno (BFGS) or Sequential Least Squares Programming (SLSQP) one needs to numerically find the function (5.24) gradient, what would be extremely computationally heavy to accomplish, given the function complexity. Moreover, such methods also do not guarantee a global solution only a local one. On the other hand, heuristic methods such as Simulated Annealing and Differential Evolution are derivative-free and global, thus are preferable than the formers.

In the following sections we present and discuss each of these methods.

5.2.1 Simulated Annealing

Simulated annealing [42], [43] is a probabilistic optimization technique adapted from the Metropolis–Hastings algorithm [44] suitable for large and complex problems. It is especially effective when the global minimum is surrounded by many local minima.

The name of the method comes from an analogy of the thermodynamical process of annealing, heating and then slowly cooling the metal. At higher temperatures, the atoms can freely move, wherewith in lower temperatures, mobility is lost. If the metal is cooled slowly, then the atoms are able to line themselves up and form a refined equilibrium structure (pure crystal). This structure is the state of minimum energy for the system. If, on the other hand, the metal is rapidly cooled, then it reaches a polycrystalline or amorphous state with higher energy [45].

To use this natural process in optimization problems, we consider the system state as the problem solution, and the energy as the objective function evaluated at the solution. Naturally, we consider the temperature as a function of the number of iterations, so that at beginning of the process we can freely search the solution space—high temperature—and as the number of iterations increases, the search space becomes narrower—low temperature. The probability distribution that the system will be in a certain state is given by the Boltzmann distribution^{III} [46]:

$$P(E) \sim e^{-\frac{E}{kT}}, \quad (5.25)$$

where E is the energy of the state, k is the Boltzmann constant and T is the temperature of the system. Equation (3) express the idea that even if the probability becomes increasingly smaller as the temperature decreases, there is a chance of the system be in a higher energy state even at low temperatures. Nonetheless, it is possible that the system changes from a local minimum state (higher energy) to a global one (lower energy).

A slight modification can be made in (5.25) by assuming that $k = 1$ and the system has changed from an energy E_1 to E_2 , from which we obtain:

$$P(E_2 - E_1) \sim e^{-\frac{E_2 - E_1}{T}}. \quad (5.26)$$

Thus, making the already mentioned analogy, we enunciate the method as follows:

1. Choose an initial solution x_0 and temperature T_0 ;
2. Randomly perturb the current solution: $x_{\text{new}} = x + \Delta x$;
3. If $f(x_{\text{new}}) < f(x)$, then accept the solution: $x = x_{\text{new}}$;
4. Else, if $\text{rand}(0, 1) < e^{-\frac{f(x_{\text{new}}) - f(x)}{T}}$, then accept the solution: $x = x_{\text{new}}$. Otherwise, reject it;

---

^{III} Also called Gibbs distribution.

5. Decrease temperature: $T = T_{\text{new}}$;
6. Repeat the process until a convergence criterion is achieved.

Due to step 4, uphill solutions can be accepted and, therefore, local minima can be escaped. To implement the method we must first decide how to perturb the solution, the initial temperature, and the *annealing schedule*—or by how much the temperature decreases at each iteration.

Choosing a good initial temperature is an important step, as it will primarily control the probability of acceptance of an uphill solution. Starting with overly small T will most likely transform the simulated annealing in a downhill search method, likewise for a large T , almost all uphill solutions will be accepted and computational time is wasted searching the global optimum in regions where it is very unlikely to be found. From (5.26), we can say that T_0 must be proportional to the distance of the local minima. In general, there are four ways of selecting T_0 :

1. Prior knowledge of the optimization problem;
2. Test several temperature values until one gives a good overall result;
3. Generate n random points $x_1, \dots, x_n$ in the search space, and choose

$$T_0 = \max_{i=1, \dots, n, j=i+1} \text{dist}(f(x_i), f(x_j));$$

4. Generate n random points $x_1, \dots, x_n$ in the search space, and choose

$$T_0 = \text{std}(\{f(x_i)\}_{i=1, \dots, n}),$$

where std is the standard deviation.

A good selection of *annealing schedule* is similarly important, as it also controls the acceptance ratio of increasing solutions. Several cooling rules can be found in literature [45], [47], the most famous ones are presented below:

1. $T = T_0 \alpha^k$, where $\alpha \in (0, 1)$;
2. $T = T_0(1 + \frac{k}{N})^\alpha$, where $N \propto$ to the maximum number of iterations, and $\alpha \in \{2, 3, 4, \dots\}$;
3. $T = \frac{T_0}{1 + \log(k+1)}$;
4. $T = \frac{T_0}{\log(k+1)}$;
5. $T = \frac{T_0}{k}$.

In all of them k is the iteration number. There is no general criterion to select one rule, as it is problem dependent. However, we verified that choosing the first rule with $\alpha = 0.95$ or 0.8 usually produces a good start.

As stated in [45], it is hard to determine Δx and each author chooses its own way. Frequently, Δx has a length proportional to T or $\sqrt{T}$ and a random direction (obtained from a uniform or Gaussian distribution).

In this work, we have employed a modified version of the method proposed in [47] and its algorithm is given as follows:

Data: $f(\cdot) \rightarrow$ objective function; $x_0 \rightarrow n$ -dimensional initial solution vector;
 $U \rightarrow n$ -dimensional upper bound vector; $L \rightarrow n$ -dimensional lower bound vector; $T_0 \rightarrow$ initial temperature;
 $N \rightarrow$ maximum number of iteration;
 $NS \rightarrow$ number of neighborhood searches; $\alpha \rightarrow$ cooling rate;
 $\epsilon \rightarrow$ relative error tolerance.

begin

$x = x_0$

$x_{\text{best}} = x_0$

$T = T_0$

for k **in** $\text{range}(N)$:

Update the temperature $T = T_0 \alpha^k$

Begin the search from the best solution $x_k = x_{\text{best}}$

for i **in** $\text{range}(NS)$:

for j **in** $\text{range}(n)$:

Create a trial solution $x_{\text{new}} = x_k$

Generate a random number $u = \text{rand}(0, 1)$

Evaluate a random perturbation $\Delta x = \text{sgn}(u - 0.5)T((1 + 1/T)^{|2u-1|} - 1)$

Perturbate the j -th component of the trial vector $x_{\text{new}}[j] = x_k[j] + \Delta x(U[j] - L[j])$

Check if the component is within its bounds and perturbate it again if it is not

while $U[j] < x_{\text{new}}[j]$ **or** $x_{\text{new}}[j] < L[j]$:

Generate a random number $u = \text{rand}(0, 1)$

Evaluate a random perturbation $\Delta x = \text{sgn}(u - 0.5)T((1 + 1/T)^{|2u-1|} - 1)$

Perturbate the j -th component of the trial vector $x_{\text{new}}[j] = x_k[j] + \Delta x(U[j] - L[j])$

Test if the trial solution is better than the current solution x_k

if $f(x_{\text{new}}) < f(x_k)$:

$x = x_{\text{new}}$

Verify if the trial solution is also better than the best solution

if $f(x_{\text{new}}) < f(x_k)$:

$x = x_{\text{new}}$

Otherwise, apply the probabilistic criterion

else:

if $\text{rand}(0, 1) < e^{-\frac{f(x_{\text{new}}) - f(x_k)}{T}}$:

$x = x_{\text{new}}$

Check if the optimization can be terminated

if $|f(x_k) - f(x_{k-s})| < \epsilon$ $s = 1, 2, 3$ **and** $|f(x_k) - f(x_{\text{best}})| < \epsilon$:

return x_{best}

return x_{best}

Algorithm 2: Simulated Annealing

5.2.2 Differential Evolution

Differential evolution [48] is also a derivative-free optimization technique. It optimizes a problem by first generating a population of possible solutions and then creating new solutions by combining the previous ones. The population member who has an optimum value for the objective function is regarded as the solution. Therefore, the objective function is only used as a measure of quality and there is no need to calculate the function gradient to perform the optimization.

The following algorithm presents an implementation for the differential evolution method [49]:

Data:

$f(\cdot) \rightarrow$ optimization function;

$N_p \rightarrow$ population size;

$\text{maxgen} \rightarrow$ maximum number of generations;

$\text{CR} \rightarrow$ crossover probability;

$F \rightarrow$ scale factor.

begin

Create a random initial population $x_i, i = 1, 2, \dots, N_p$

Find $x_{\text{best}} = \arg \max_i f(x_i)$

for k **in** $\text{range}(\text{maxgen})$:

for i **in** $\text{range}(N_p)$:

    Randomly pick two members of the population x_{r0} and x_{r1} , such

$x_i \neq x_{r0} \neq x_{r1}$

    Choose a random index $j_r = \text{randint}(0, n)$

    Create a crossover vector $c[j] = \text{rand}(0, 1) \ j = 1, 2, \dots, n$

for j **in** $\text{range}(n)$:

      Construct a trial candidate x_t as follows:

if $c[j] < \text{CR}$ **or** $j = j_r$:

$x_t[j] = x_{\text{best}}[j] + F(x_{r0}[j] - x_{r1}[j])$

else:

$x_t[j] = x_i[j]$

if $f(x_t) < f(x_i)$:

$x_i = x_t$

if $f(x_t) < f(x_{\text{best}})$:

$x_{\text{best}} = x_t$

return x_{best}

Algorithm 3: Differential Evolution

Algorithm 3 has four parameters: N_p controls the population size and is usually chosen to be ten times the size of the solution vector; maxgen controls the number of generations; $F \in \mathbb{R}_{>0}$ controls the rate at which the population evolves; and $\text{CR} \in [0, 1]$ controls the ratio of components values are copied from the mutant. Selecting $F \in [0.5, 1]$ randomly—known in literature as dither technique—at each generation or trial vector improves the method convergence [50].

Literature [49] refers to two main ways of doing the crossover. The first one is called exponential crossover: One component of the mutant vector is randomly chosen and copied to the correspondent trial vector component. The subsequent parameters are determined by comparing CR to a newly generated uniformly distributed random number between 0 and 1, or $\text{rand}_j(0, 1)$. Providing that $\text{rand}_j(0, 1) < \text{CR}$, mutant components will be copied to the trial vector, but the first time $\text{rand}_j(0, 1) \geq \text{CR}$, the current and all remaining components will be taken from the original vector. The exponential name comes from the fact that the number of inherited mutant parameters is an exponentially distributed random variable. The second one is called binomial crossover. As with the exponential, it compares CR to $\text{rand}_j(0, 1)$ and then decides if the trial component will come from either the original or the mutant vector. If $\text{rand}_j(0, 1) < \text{CR}$ the j -th trial component is copied from the j -th mutant one; otherwise is copied from the j -th original one. Likewise, the binomial name comes from the fact that the number of inherited mutant parameters is a binomially distributed random variable.

There are several variants on how to obtain the mutant vector [50]. We present some in the table below.

Table 1: Mutation Variants

Nomenclature	Variant
rand1	$x_t = x_{r_2} + F(x_{r_0} - x_{r_1})$, where $x_i \neq x_{r_0} \neq x_{r_1} \neq x_{r_2}$
best1	$x_t = x_{\text{best}} + F(x_{r_0} - x_{r_1})$, where $x_i \neq x_{r_0} \neq x_{r_1}$
rand2	$x_t = x_{r_4} + F(x_{r_0} - x_{r_1} + x_{r_2} - x_{r_3})$, where $x_i \neq x_{r_0} \neq x_{r_1} \neq x_{r_2} \neq x_{r_3} \neq x_{r_4}$
best2	$x_t = x_{\text{best}} + F(x_{r_0} - x_{r_1} + x_{r_2} - x_{r_3})$, where $x_i \neq x_{r_0} \neq x_{r_1} \neq x_{r_2} \neq x_{r_3}$
randtobest1	$x_t = x_{r_0} + F(x_{\text{best}} - x_{r_0}) + F(x_{r_1} - x_{r_2})$, where $x_i \neq x_{r_0} \neq x_{r_1} \neq x_{r_2}$
currenttobest1	$x_t = x_i + F(x_{\text{best}} - x_i) + F(x_{r_0} - x_{r_1})$, where $x_i \neq x_{r_0} \neq x_{r_1}$
currenttorand1	$x_t = x_i + F(x_{r_2} - x_i) + F(x_{r_0} - x_{r_1})$, where $x_i \neq x_{r_0} \neq x_{r_1} \neq x_{r_2}$

By combining both mutation and crossover options, we obtain the following strategies as depicted in Table 2.

Table 2: Differential Evolution Strategies

Differential Evolution Strategy
best1bin
best1exp
rand1bin
rand1exp
rand2bin
rand2exp
best2bin
best2exp
randtobest1bin
randtobest1exp
currenttobest1bin
currenttobest1exp
currenttorand1bin
currenttorand1exp

Algorithm 3 follows the best1bin strategy.

Several termination/convergence criteria can be chosen:

1. Generation based: Terminate the optimization process if the number of generations exceeds maxgen;
2. Time based: Terminate the optimization process if the execution time surpasses a predefined $t_{\max}$;
3. Classical: Terminate the optimization if after m generations:
  - a) $|f(x_{\text{best}}^k) - f(x_{\text{best}}^{k-m})| < \epsilon_f$, where x_{best}^k is the best solution at the k -th generation;
  - b) $\|x_{\text{best}}^k - x_{\text{best}}^{k-m}\| < \epsilon_x$, where x_{best}^k is the best solution at the k -th generation;
4. Population statistics based: Terminate the optimization process if:
  - a) Standard deviation of the populations is below a threshold ϵ_{std} or $\text{std}(\text{pop}) < \epsilon$;
  - b) Standard deviation of the population is below $\epsilon_{\text{std}} + \epsilon_{\mu} \overline{\text{pop}}$ or $\text{std}(\text{pop}) < \epsilon_{\text{std}} + \epsilon_{\mu} |f(\text{pop})|$ [51], where $\overline{f(\text{pop})}$ is the mean of the objective function applied to the population;
5. Distance based: Terminate the optimization process if $\max_i \text{dist}(x_i, x_{\text{best}}) < \epsilon_d$.

5.2.3 Testing the Algorithms

To exemplify the use of both algorithms, Simulated Annealing and Differential Evolution, consider the following optimization problem known in literature as Eggholder

function [52]:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) = -x_1 \sin\left(\sqrt{|x_1 - x_2 - 47|}\right) - (x_2 + 47) \sin\left(\sqrt{\left|\frac{1}{2}x_1 + x_2 + 47\right|}\right) \\ \text{subject to} \quad & -512 \leq x_i \leq 512, \quad i = 1, 2. \end{aligned} \quad (5.27)$$

Figure 21 depicts the graph of function (5.27). This function exhibits several extrema, thus it provides a challenge to conventional optimization methods. The global minimum of (5.27) is:

$$f(\mathbf{x}^*) = -959.6407, \quad \text{at } \mathbf{x}^* = [512, 404.2319]^\top \quad (5.28)$$

Figure 21: Eggholder Function

Applying the Simulated Annealing method with $x_0 = [-461.305, 338.518]^\top$; $T_0 = 300$; $N = 200$; $NS = 20$; $\alpha = 0.8$; and $\epsilon = 1e-06$, we obtained a global minimum of $f(\mathbf{x}^*) = -959.6406$ at $\mathbf{x}^* = [511.9999, 404.2318]^\top$.

Figure 22 shows the best solution evolution together with the level curves of (5.27). We can see that the algorithm is able to escape from the local minimum located on the top left and successfully find the global minimum on the top right.

Figure 22: Best Solution Evolution - Simulated Annealing

On the other hand, applying the Differential Evolution method with: $N_p = 20$; $\text{maxgen} = 100$; $\text{CR} = 0.9$; and $F = 0.8$, yields the following best solution trajectory depicted in Figure 23.

Figure 23: Best Solution Evolution - Differential Evolution

The global minimum obtained was $f(\mathbf{x}^*) = -959.6406$ at $\mathbf{x}^* = [511.9999, 404.2318]^T$; therefore the optimization process was successful.

This example confirms that the optimization methods explored in this section can find, without calculating the derivatives, the global minimum of complicated functions. They are, consequently, a perfect fit for the proposed optimization problem. In the next chapter, we shall also see that they outperform the Monte Carlo method.

5.3 Sensitivity

In this section, we discuss a methodology to find the most influential power system parameters to the transient analysis assessment problem. Identified such parameters, we reduce the number of uncertain parameters to only the influential ones. This selection can provide a significant variation in the assessment speed of larger power systems and can be regarded as a crucial step to the transient analysis of power systems with uncertainties. We start by considering a power system modeled by the following initial value problem:

$$\dot{x} = f(x, \lambda), \quad x(t_0) = x_0, \quad (5.29)$$

where $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$ are, respectively, the system state and parameter vectors. The solution of (5.29) is represented by:

$$x(t, \lambda). \quad (5.30)$$

To analyze the effect of the parameters in (5.29), let us add $\Delta\lambda_k$ to the k -th parameter λ_k . Since λ is a vector, we adopt the following notation:

$$\zeta_k = \begin{bmatrix} 0 \\ \vdots \\ \Delta\lambda_k \\ \vdots \\ 0 \end{bmatrix}, \quad (5.31)$$

where ζ_k is a perturbation vector. The Taylor series of the perturbed solution is as follows:

$$x(t, \lambda + \zeta_k) = \sum_{j=0}^{\infty} \left[\frac{1}{j!} \left(\Delta\lambda_k \frac{\partial}{\partial \lambda_k} \right)^j x(t, \lambda) \right]. \quad (5.32)$$

Truncating (5.32) in the first term yields:

$$x(t, \lambda + \zeta_k) - x(t, \lambda) \approx \frac{\partial x(t, \lambda)}{\partial \lambda_k} \Delta\lambda_k \rightarrow \Delta x \approx s_k \Delta\lambda_k, \quad (5.33)$$

where s_k is called the local sensitivity of x with respect to the parameter λ_k . Naturally, the local sensitivity of x_i with respect to the parameter λ_k is:

$$s_{ik} = \frac{\partial x_i(t, \lambda)}{\partial \lambda_k}. \quad (5.34)$$

The sensitivity matrix is then given by:

$$S = (s_{ik}), \quad (5.35)$$

where $i = 1, 2, \dots, n$ and $k = 1, 2, \dots, m$.

Because (5.34) is not dimensionless, we define a normalized sensitivity as follows:

$$\tilde{s}_{ik} = \frac{\lambda_k}{x_i} \frac{\partial x_i(t, \lambda)}{\partial \lambda_k} = \frac{\partial \ln(x_i(t, \lambda))}{\partial \ln(\lambda_k)}. \quad (5.36)$$

Likewise, the normalized sensitivity matrix is given by:

$$\tilde{S} = (\tilde{s}_{ik}), \quad (5.37)$$

where $i = 1, 2, \dots, n$ and $k = 1, 2, \dots, m$.

5.3.1 Approximating the Derivative

To estimate the value of (5.34), we employ the Taylor series of a function $f(x)$ as follows:

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{1}{2!} \Delta x^2 f''(x) + \frac{1}{3!} \Delta x^3 f'''(x) + \mathcal{O}(\Delta x^4). \quad (5.38)$$

Choosing Δx to be h yields:

$$f(x + h) = f(x) + h f'(x) + \mathcal{O}(h^2) \rightarrow f'(x) = \frac{f(x + h) - f(x)}{h} - \frac{\mathcal{O}(h^2)}{h}. \quad (5.39)$$

Similarly, choosing Δx to be $-h$ results in:

$$f(x - h) = f(x) - h f'(x) + \mathcal{O}(h^2) \rightarrow f'(x) = \frac{f(x) - f(x - h)}{h} + \frac{\mathcal{O}(h^2)}{h}. \quad (5.40)$$

Equation (5.39) are known, respectively, as forward and backward difference formula for approximating the derivative of a function. The error can be estimated using the Lagrange remainder:

$$R_n = \frac{f^{(n+1)}(\mathfrak{B})}{(n+1)!} h^{n+1}, \quad (5.41)$$

where $\mathfrak{B} \in (x, x + \Delta x)$. Then the error for each formula is respectively:

$$\mp \frac{R_2}{h} = \mp \frac{h}{2} f''(\mathfrak{B}), \quad \mathfrak{B} \in (x, x \pm h). \quad (5.42)$$

A better derivative estimate can be obtained by the following procedure. Let $\Delta x = \pm h$, then:

$$f(x + h) = f(x) + h f'(x) + \frac{h^2}{2} f''(x) + \mathcal{O}(h^3), \quad (5.43)$$

$$f(x - h) = f(x) - h f'(x) + \frac{h^2}{2} f''(x) - \mathcal{O}(h^3). \quad (5.44)$$

Subtracting (5.44) from (5.43) yields:

$$f(x+h) - f(x-h) = 2hf'(x) + 2\mathcal{O}(h^3) \rightarrow f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{\mathcal{O}(h^3)}{h}. \quad (5.45)$$

Equation (5.45) is known as central difference formula. The error can be estimated as:

$$-\frac{1}{2h}(R_{3_1} - R_{3_2}) = -\frac{1}{2h} \left(\frac{f'''(\mathfrak{B}_1)}{3!}h^3 - \frac{f'''(\mathfrak{B}_2)}{3!}(-h)^3 \right) = -\frac{1}{2h} \left(\frac{f'''(\mathfrak{B}_1)}{3!}h^3 + \frac{f'''(\mathfrak{B}_2)}{3!}h^3 \right), \quad (5.46)$$

where $\mathfrak{B}_1 \in (x, x+h)$ and $\mathfrak{B}_2 \in (x-h, x)$. By the intermediate value theorem, we obtain:

$$f'''(\mathfrak{B}) = \frac{1}{2}(f'''(\mathfrak{B}_1) + f'''(\mathfrak{B}_2)), \quad \mathfrak{B} = (x-h, x+h). \quad (5.47)$$

Thus, the error is:

$$-\frac{1}{2h} \left(2 \frac{f'''(\mathfrak{B})}{6} h^3 \right) \rightarrow -\frac{h^2}{6} f'''(\mathfrak{B}), \quad \mathfrak{B} = (x-h, x+h). \quad (5.48)$$

Using only forward or backward values, we can also obtain a better estimate. To do so, consider the following Taylor series:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \mathcal{O}(h^3), \quad (5.49)$$

$$f(x+2h) = f(x) + 2hf'(x) + \frac{4h^2}{2}f''(x) + \mathcal{O}(8h^3). \quad (5.50)$$

Subtracting (5.49) from (5.50) yields:

$$\begin{aligned} f(x+2h) - 4f(x+h) &= -3f(x) - 2hf'(x) + \mathcal{O}(4h^3) \rightarrow \\ f'(x) &= \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + \frac{\mathcal{O}(4h^3)}{2h}. \end{aligned} \quad (5.51)$$

Equation (5.51) is known as three point endpoint formula. The error can be estimated as follows:

$$\frac{1}{2h}(-4R_{3_1} + R_{3_2}) = \frac{1}{2h} \left(-4 \frac{f'''(\mathfrak{B}_1)}{3!}h^3 + \frac{f'''(\mathfrak{B}_2)}{3!}(2h)^3 \right) = \frac{1}{2h} \left(-4 \frac{f'''(\mathfrak{B}_1)}{3!}h^3 + 8 \frac{f'''(\mathfrak{B}_2)}{3!}h^3 \right), \quad (5.52)$$

where $\mathfrak{B}_1 \in (x, x+h)$ and $\mathfrak{B}_2 \in (x, x+2h)$. Using the intermediate value theorem, we obtain:

$$f'''(\mathfrak{B}) = \frac{1}{4}(-4f'''(\mathfrak{B}_1) + 8f'''(\mathfrak{B}_2)), \quad \mathfrak{B} = (x, x+2h). \quad (5.53)$$

Finally, the error is:

$$\frac{1}{2h} \left(4 \frac{f'''(\mathfrak{B})}{6} h^3 \right) \rightarrow \frac{h^2}{3} f'''(\mathfrak{B}), \quad \mathfrak{B} = (x, x+2h). \quad (5.54)$$

Following the same procedure for $-h$, we arrive in the next formula:

$$f'(x) = \frac{f(x-2h) - 4f(x-h) + 3f(x)}{2h} + \frac{h^2}{3} f'''(\mathfrak{B}), \quad \mathfrak{B} = (x-2h, x). \quad (5.55)$$

5.3.2 PCA - Principal Component Analysis

Until now, we dealt only with a single parameter variation. To quantify the effect of several parameters changing at the same time, consider the following objective function [53]:

$$J(p) = \int_0^t \sum_{i=1}^n \left(\frac{x_i(p, \tau) - x_i(p_0, \tau)}{x_i(p_0, \tau)} \right)^2 d\tau, \quad (5.56)$$

where $x_i(p, t)$ and $x_i(p_0, t)$ are, respectively, the perturbed and nominal solutions; and $p = \ln(\lambda)$. Assuming that both solutions are sampled in l points, we obtain:

$$J(p) = \int_0^t \sum_{i=1}^n \sum_{j=0}^{l-1} \left(\frac{x_i(p, \tau) - x_i(p_0, \tau)}{x_i(p_0, \tau)} \right)^2 \delta(\tau - j) d\tau = \sum_{j=0}^{l-1} \sum_{i=1}^n \left[\frac{x_i(p, t_j) - x_i(p_0, t_j)}{x_i(p_0, t_j)} \right]^2, \quad (5.57)$$

where $\delta(\cdot)$ is the Dirac delta function.

The Taylor series for $J(p)$ centered at p_0 is as follows:

$$J(p) = J(p_0) + \nabla J(p_0)(p - p_0) + \frac{1}{2}(p - p_0)^T \nabla^2 J(p_0)(p - p_0) + \dots \quad (5.58)$$

Since $J(p_0) = 0$ and is a minimum, which implies in $\nabla J(p_0) = 0$, we have:

$$J(p) \approx \frac{1}{2}(p - p_0)^T \nabla^2 J(p_0)(p - p_0). \quad (5.59)$$

To find the Hessian matrix expression, we proceed as follows:

$$J(p) = \sum_{j=0}^{l-1} \sum_{i=1}^n \left[\frac{r_i(t_j)}{x_i(p_0, t_j)} \right]^2 \rightarrow \nabla J(p)_k = \sum_{j=0}^{l-1} \sum_{i=1}^n \frac{\partial}{\partial p_k} \left(\left[\frac{r_i(t_j)}{x_i(p_0, t_j)} \right]^2 \right), \quad (5.60)$$

where $r_i(t_j)$ is the i -th residue at a time t_j . Applying the chain rule in (5.60) yields:

$$\nabla J(p)_k = 2 \sum_{j=0}^{l-1} \sum_{i=1}^n \frac{r_i(t_j)}{x_i(p_0, t_j)} \frac{\partial}{\partial p_k} \left(\frac{r_i(t_j)}{x_i(p_0, t_j)} \right). \quad (5.61)$$

Then,

$$\begin{aligned} \nabla^2 J(p)_{ks} &= \frac{\partial}{\partial p_s} 2 \left[\sum_{j=0}^{l-1} \sum_{i=1}^n \frac{r_i(t_j)}{x_i(p_0, t_j)} \frac{\partial}{\partial p_k} \left(\frac{r_i(t_j)}{x_i(p_0, t_j)} \right) \right] \rightarrow \\ &2 \sum_{j=0}^{l-1} \sum_{i=1}^n \left[\frac{\partial}{\partial p_s} \left(\frac{r_i(t_j)}{x_i(p_0, t_j)} \right) \frac{\partial}{\partial p_k} \left(\frac{r_i(t_j)}{x_i(p_0, t_j)} \right) + \frac{r_i(t_j)}{x_i(p_0, t_j)} \frac{\partial^2}{\partial p_s \partial p_k} \left(\frac{r_i(t_j)}{x_i(p_0, t_j)} \right) \right]. \end{aligned} \quad (5.62)$$

Using Gauss approximation on (5.62) results in:

$$\nabla^2 J(p)_{ks} \approx 2 \sum_{j=0}^{l-1} \sum_{i=1}^n \frac{\partial}{\partial p_s} \left(\frac{r_i(t_j)}{x_i(p_0, t_j)} \right) \frac{\partial}{\partial p_k} \left(\frac{r_i(t_j)}{x_i(p_0, t_j)} \right). \quad (5.63)$$

Because $r_i(t_j) = x_i(p, t_j) - x_i(p_0, t_j)$, we have:

$$\frac{\partial}{\partial p_k} \left(\frac{r_i(t_j)}{x_i(p_0, t_j)} \right) = \frac{\partial}{\partial p_k} \left(\frac{x_i(p, t_j)}{x_i(p_0, t_j)} - 1 \right) = \frac{1}{x_i(p_0, t_j)} \frac{\partial x_i(p, t_j)}{\partial p_k} \xrightarrow[p=p_0]{\text{at}} \frac{\partial \ln(x_i(p_0, t_j))}{\partial p_k}. \quad (5.64)$$

Thus,

$$\nabla^2 J(p_0) = 2 \sum_{j=0}^{l-1} \tilde{S}_j^\top \tilde{S}_j \rightarrow J(p) \approx (p - p_0)^\top \sum_{j=0}^{l-1} \tilde{S}_j^\top \tilde{S}_j (p - p_0), \quad (5.65)$$

where $\tilde{S}_j$ is the normalized sensitivity matrix at t_j .

To understand how $J(p)$ varies with p , we can use the eigendecomposition of $\sum_{j=0}^{l-1} \tilde{S}_j^\top \tilde{S}_j$ as follows [4]:

$$\sum_{j=0}^{l-1} \tilde{S}_j^\top \tilde{S}_j = U \Pi U^\top, \quad (5.66)$$

where Π is the diagonal matrix of the eigenvalues π_i of $\sum_{j=0}^{l-1} \tilde{S}_j^\top \tilde{S}_j$ and U is the square matrix whose i -th column is the u_i normalized eigenvector of $\sum_{j=0}^{l-1} \tilde{S}_j^\top \tilde{S}_j$. Therefore, $J(p)$ is most sensitive to changes in p along the eigenvector corresponding to the largest eigenvalue. As such, literature refers to $\rho = U^\top(p - p_0)$ as the principal components.

The most influential parameters can be identified as follows: let $\rho_{\max}$ be the principal component corresponding to the largest eigenvalue, $\pi_{\max}$, of $\sum_{j=0}^{l-1} \tilde{S}_j^\top \tilde{S}_j$, then:

$$\rho_{\max} = u_{\max}^\top (p - p_0), \quad (5.67)$$

where $u_{\max}$ is the eigenvector corresponding to $\pi_{\max}$. Since $u_{\max}$ is unitary, it gives the percentage influence of the variation of each parameter. We can then enumerate the most influential parameters and select those who will be included in a subsequent analysis by choosing a predefined threshold percentage value and comparing them to it.

Clearly, for a large system it is unpractical to calculate all eigenvalues. Luckily, as only the largest eigenvalue and its correspondent eigenvector are of our interest, we can employ fast methods such as Power Iteration or Lanczos algorithm^{IV} to find them.

5.3.3 Analysis of the Sensitivity Matrix

Another way to identify the most influential parameters, is by analyzing directly the sensitivity matrix. To do so, let $\tilde{S}$ be the sensitivity matrix at a time t , then:

$$\tilde{S} = \begin{bmatrix} \tilde{s}_{11} & \cdots & \tilde{s}_{1m} \\ \vdots & \ddots & \vdots \\ \tilde{s}_{n1} & \cdots & \tilde{s}_{nm} \end{bmatrix}. \quad (5.68)$$

The $L_{2,1}$ norm of $\tilde{S}$ is defined by:

$$\|\tilde{S}\|_{2,1} = \sum_{j=1}^m \left(\sum_{i=1}^n \tilde{s}_{ij}^2 \right)^{1/2}. \quad (5.69)$$

Equation (5.69) shows that the matrix norm, as defined, is composed by the sum of the “norm” of each column. Thus, we may say that the most influential parameters will be

^{IV} Since $\sum_{j=0}^{l-1} \tilde{S}_j^\top \tilde{S}_j$ is symmetric.

those whose columns most contribute to the overall norm. In practical terms, we are not considering the effect of several parameters varying and assuming that the influence can be estimated as being the norm of the sensitivity of x with respect to each parameter λ . Differently from the Principal Component Analysis, this method does not give a percentage influence and it is not normalized. We cannot easily select a predefined threshold value and select the most dominant parameters. Of course, for a small system, we can manually select those parameters, but for a large system is impractical. Hence, we propose the use of a clustering technique called K-Means [5] to divide the parameters in two clusters: most sensitive parameters and least sensitive parameters. In the following subsection we will briefly introduce the K-Means technique.

5.3.4 K-Means

K-Means is a simple clustering algorithm which cluster the data in K clusters. The main objective of the algorithm is to maximize the variance of all clusters in respect to the each other, while minimizing the variance within each cluster. The steps for executing the algorithm, assuming that we have $X = x_1, x_2, \dots, x_n$ data vectors, are as follows:

1. Choose the number of clusters, K ;
2. Randomly select K cluster centroids: $C = c_1, c_2, \dots, c_K$;
3. Assign each data vector x_i to the closest centroid c_j , i.e.,

$$\xi_j = \arg \min_{x \in X} d(c_j, x), \quad (5.70)$$

where ξ_j is the set of all points assigned to the j -th cluster and $d(\cdot)$ is a distance (under some metric) function.

4. Find the new cluster centroids as the mean of all points assigned to its cluster:

$$c_i = \frac{\sum_{x_i \in \xi_i} x_i}{|\xi_i|}; \quad (5.71)$$

5. Repeat steps 3 and 4 for a predetermined number of iterations or until the cluster centroids vary very little between iterations.

An easy way to assign each data vector to its corresponding cluster is by defining a binary matrix $\mathbf{f}$, where:

$$\mathbf{f}_{ij} = \begin{cases} 1 & \text{if } d(c_j, x_i) = \min_{x \in X} d(c_j, x), \\ 0 & \text{otherwise.} \end{cases} \quad (5.72)$$

Using matrix $\mathbf{f}$, equation (5.71) can be modified accordingly:

$$c_j = \frac{\sum_i \mathbf{f}_{ij} x_i}{\sum_i \mathbf{f}_{ij}}. \quad (5.73)$$

5.3.5 Power System Parameter Sensitivity

Let us analyze only the machine angles—as they dominate^V over t_{cr} . For a general power system with m loads, l lines, and n machines:

$$\delta = [\delta_1, \delta_2, \dots, \delta_n]^\top \rightarrow \text{Machine angles}, \quad (5.74)$$

$$P_L = [P_{L_1}, P_{L_2}, \dots, P_{L_m}]^\top \rightarrow \text{Active loads}, \quad (5.75)$$

$$Q_L = [Q_{L_1}, Q_{L_2}, \dots, Q_{L_m}]^\top \rightarrow \text{Reactive loads}, \quad (5.76)$$

$$R = [R_1, R_2, \dots, R_l]^\top \rightarrow \text{Line resistances}, \quad (5.77)$$

$$X = [X_1, X_2, \dots, X_l]^\top \rightarrow \text{Line reactances}, \quad (5.78)$$

$$B/2 = [B_1, B_2, \dots, B_l]^\top \rightarrow \text{Shunts}, \quad (5.79)$$

and

$$S = \begin{bmatrix} \frac{\partial \delta_1}{\partial P_{L_1}} & \dots & \frac{\partial \delta_1}{\partial P_{L_m}} & \left| \frac{\partial \delta_1}{\partial Q_{L_1}} & \dots & \frac{\partial \delta_1}{\partial Q_{L_m}} \right| & \frac{\partial \delta_1}{\partial R_1} & \dots & \frac{\partial \delta_1}{\partial R_l} & \left| \frac{\partial \delta_1}{\partial X_1} & \dots & \frac{\partial \delta_1}{\partial X_l} \right| & \frac{\partial \delta_1}{\partial B/2_1} & \dots & \frac{\partial \delta_1}{\partial B/2_l} \\ \frac{\partial \delta_2}{\partial P_{L_1}} & \dots & \frac{\partial \delta_2}{\partial P_{L_m}} & \left| \frac{\partial \delta_2}{\partial Q_{L_1}} & \dots & \frac{\partial \delta_2}{\partial Q_{L_m}} \right| & \frac{\partial \delta_2}{\partial R_1} & \dots & \frac{\partial \delta_2}{\partial R_l} & \left| \frac{\partial \delta_2}{\partial X_1} & \dots & \frac{\partial \delta_2}{\partial X_l} \right| & \frac{\partial \delta_2}{\partial B/2_1} & \dots & \frac{\partial \delta_2}{\partial B/2_l} \\ \vdots & \dots & \vdots & \left| \vdots & \dots & \vdots \right| & \vdots & \dots & \vdots & \left| \vdots & \dots & \vdots \right| & \vdots & \dots & \vdots \\ \frac{\partial \delta_n}{\partial P_{L_1}} & \dots & \frac{\partial \delta_n}{\partial P_{L_m}} & \left| \frac{\partial \delta_n}{\partial Q_{L_1}} & \dots & \frac{\partial \delta_n}{\partial Q_{L_m}} \right| & \frac{\partial \delta_n}{\partial R_1} & \dots & \frac{\partial \delta_n}{\partial R_l} & \left| \frac{\partial \delta_n}{\partial X_1} & \dots & \frac{\partial \delta_n}{\partial X_l} \right| & \frac{\partial \delta_n}{\partial B/2_1} & \dots & \frac{\partial \delta_n}{\partial B/2_l} \end{bmatrix}. \quad (5.80)$$

From (5.80), we find $\tilde{S}$ as follows:

$$\tilde{S} = \begin{bmatrix} \frac{P_{L_1}}{\delta_1} \frac{\partial \delta_1}{\partial P_{L_1}} & \dots & \frac{P_{L_m}}{\delta_1} \frac{\partial \delta_1}{\partial P_{L_m}} & \left| \frac{Q_{L_1}}{\delta_1} \frac{\partial \delta_1}{\partial Q_{L_1}} & \dots & \frac{Q_{L_m}}{\delta_1} \frac{\partial \delta_1}{\partial Q_{L_m}} \right| & \frac{R_1}{\delta_1} \frac{\partial \delta_1}{\partial R_1} & \dots & \frac{R_l}{\delta_1} \frac{\partial \delta_1}{\partial R_l} & \left| \frac{X_1}{\delta_1} \frac{\partial \delta_1}{\partial X_1} & \dots & \frac{X_l}{\delta_1} \frac{\partial \delta_1}{\partial X_l} \right| & \frac{B/2_1}{\delta_1} \frac{\partial \delta_1}{\partial B/2_1} & \dots & \frac{B/2_l}{\delta_1} \frac{\partial \delta_1}{\partial B/2_l} \\ \frac{P_{L_1}}{\delta_2} \frac{\partial \delta_2}{\partial P_{L_1}} & \dots & \frac{P_{L_m}}{\delta_2} \frac{\partial \delta_2}{\partial P_{L_m}} & \left| \frac{Q_{L_1}}{\delta_2} \frac{\partial \delta_2}{\partial Q_{L_1}} & \dots & \frac{Q_{L_m}}{\delta_2} \frac{\partial \delta_2}{\partial Q_{L_m}} \right| & \frac{R_1}{\delta_2} \frac{\partial \delta_2}{\partial R_1} & \dots & \frac{R_l}{\delta_2} \frac{\partial \delta_2}{\partial R_l} & \left| \frac{X_1}{\delta_2} \frac{\partial \delta_2}{\partial X_1} & \dots & \frac{X_l}{\delta_2} \frac{\partial \delta_2}{\partial X_l} \right| & \frac{B/2_1}{\delta_2} \frac{\partial \delta_2}{\partial B/2_1} & \dots & \frac{B/2_l}{\delta_2} \frac{\partial \delta_2}{\partial B/2_l} \\ \vdots & \dots & \vdots & \left| \vdots & \dots & \vdots \right| & \vdots & \dots & \vdots & \left| \vdots & \dots & \vdots \right| & \vdots & \dots & \vdots \\ \frac{P_{L_1}}{\delta_n} \frac{\partial \delta_n}{\partial P_{L_1}} & \dots & \frac{P_{L_m}}{\delta_n} \frac{\partial \delta_n}{\partial P_{L_m}} & \left| \frac{Q_{L_1}}{\delta_n} \frac{\partial \delta_n}{\partial Q_{L_1}} & \dots & \frac{Q_{L_m}}{\delta_n} \frac{\partial \delta_n}{\partial Q_{L_m}} \right| & \frac{R_1}{\delta_n} \frac{\partial \delta_n}{\partial R_1} & \dots & \frac{R_l}{\delta_n} \frac{\partial \delta_n}{\partial R_l} & \left| \frac{X_1}{\delta_n} \frac{\partial \delta_n}{\partial X_1} & \dots & \frac{X_l}{\delta_n} \frac{\partial \delta_n}{\partial X_l} \right| & \frac{B/2_1}{\delta_n} \frac{\partial \delta_n}{\partial B/2_1} & \dots & \frac{B/2_l}{\delta_n} \frac{\partial \delta_n}{\partial B/2_l} \end{bmatrix}. \quad (5.81)$$

Finally, because the critical clearing time is primarily controlled by the fault-on power system behavior, we evaluate the sensitivity matrix over the fault-on trajectory. If we are employing the second methodology to identify the most influential parameters, then we only evaluate the sensitivity matrix only at the end of the trajectory (calculated over a preselected fault-on time), whereas in the first we calculate the sensitivity matrix at each point of the trajectory. The number of points will be determined by the numerical method used to solve the differential equations of the fault-on power system. Figures 24 and 25 illustrate, respectively, each methodology flowchart.

^V This consideration is important as it halves our calculations.

Figure 24: Flowchart - Principal Component Analysis

Figure 25: Flowchart - Sensitivity Matrix Analysis

6 RESULTS

In this chapter, we study the transient stability of two power systems with uncertainties: a single machine infinite bus system, and the IEEE 14 bus system [54]. These two systems allow us to verify and test all the methods that we proposed in this work: the robust PEBS, the optimization techniques, and the sensitivity analysis.

In the single machine system, in particular, we study the case where only the post-fault parameters have uncertainties. This example showcases the use of subsection 5.1.2 method. Given the simplicity of the example, we purposely skip the optimization methods and sensitivity analysis in this example. A high-precision Monte Carlo simulation can be performed in seconds, making the use of any approximation unjustifiable.

Conversely, in the IEEE 14 bus system—a multi-machine power system, we study the case where the system parameters have uncertainties in all periods. This example showcases the usage of robust PEBS and optimization methods. From the results, it will be evident to the reader that the optimization methods offer a result close to the one obtained via the Monte Carlo method^I, and in shorter assessment time. The reader will also be able to see that the robust PEBS method gives a good estimate for the critical clearing time in the shortest simulation time. A sensitivity analysis, as discussed in section 5.3, is also performed to screen the power system most influential parameters for a transient stability assessment. The results will demonstrate that the assessment with uncertainties only in the most influential parameters renders a faster analysis with an insignificant precision loss.

All programming routines were implemented in Python using the packages Scipy [51] and PyInterval [55]. In addition, all simulations were executed in a AMD Ryzen™ 7 1700 processor.

6.1 Single-machine

In this section, we perform a transient stability assessment of a single machine infinite bus system using both the robust PEBS and Monte Carlo methods. The system is depicted in Figure 26 and its parameters are listed in Table 3.

---

^I In the Appendix, we discuss how we estimated the number of iterations for the Monte Carlo method.

Figure 26: Single Machine Infinite Bus System

Table 3: System Parameters

Parameter	Pre-Fault	Fault-On	Post-Fault
$E(\text{pu})$	1	1	1
$E_\infty(\text{pu})$	1	1	1
$P_m(\text{pu})$	1	1	0.9 ± 0.1
$M(\text{pu})$	0.05	0.05	0.05
$D(\text{pu})$	0.03	0.03	0.03
$B(\text{pu})$	-1.667	-0.645	-1.176 ± 0.058
$G(\text{pu})$	0.2	0	0.2 ± 0.01

Robust PEBS provides the curves shown in Figure 27. We can observe that the critical energy V_{cr} is the first maximum of $a_p(x)$ and the critical clearing time t_{cr} estimate is obtained as the time when $b(x)$ reaches V_{cr} .

Figure 27: Energy Curves of Robust PEBS – Fault-on Trajectory

The critical clearing time estimate, obtained as the instant when the total energy $b(x)$ reaches the first maximum of the minimum potential energy $a_p(x)$, was found to be 204 ms with an execution time of 3.228 s. Conversely, the Monte Carlo method showed that the real critical clearing time is 266 ms with an execution time of 314.347 s. Hence, the proposed methodology provided a good estimate of t_{cr} with a fraction of the execution time.

6.2 Multi-Machine

In this section's first part, the robust PEBS, optimization and Monte Carlo methods are applied to assess the transient stability of the IEEE 14 bus system—illustrated in Figure 28—considering uncertainties. In the second part, a sensitivity analysis is employed in the 14 bus system to identify the most influential parameter for the transient stability assessment. In the sequence, the Monte Carlo method is used to estimate the critical clearing time of the power system two times: one with uncertainties in all parameters and another with uncertainties only in the most influential.

Figure 28: IEEE 14 bus system diagram

Source: Retrieved from <http://bit.do/14bus> [54]

For the Simulated Annealing algorithm we did not evaluate the initial temperature at each simulation, as it was time consuming. Instead we selected the initial temperature $T_0 = 150e-3$ and choose $\epsilon = 1e-6$. For the Differential evolution algorithm, we selected the rand1bin strategy, a population size 10 times the number of uncertainties and the terminating criteria is the 4.b) presented in chapter 5 with $\epsilon_\mu = 1e-3$.

6.2.1 Methodologies Benchmark

This subsection presents the transient stability assessment for the IEEE 14 bus system in three different cases:

1. A small set of transmission line parameters with uncertainties;
2. A large set of transmission line parameters with uncertainties;
3. A small set of loads with uncertainties.

These cases were purposely chosen to demonstrate that both optimization algorithms and the robust PEBS method outperform the Monte Carlo method in several situations.

6.2.1.1 Case 1

Uncertainties as depicted in Table 4 and a solid three-phase short-circuit occurs at bus 1 and the fault is cleared by the opening of the line 1-5.

Table 4: Uncertainties Case 1

Line		Parameter Uncertainty (%)		
From Bus	To Bus	R	X	B/2
1	2	10	10	1
2	4	0.1	0	0
3	4	1	0	0
6	13	1	0	0
9	10	1	0	0
12	13	0.1	0	0

For the first case we obtained the following results:

Table 5: Case 1 Results

Method	Critical Clearing Time (ms)
Robust PEBS	296
Simulated Annealing	299
Differential Evolution	292
Monte Carlo	299

Looking at Table 5, we can verify that our proposed methodology found a close result without resorting to the computationally demanding Monte Carlo methodology. We also note that the optimization methods also provided comparable results. They, however, performed slower than the robust PEBS.

Table 6: Case 1 Execution Time

Method	Time (min)
Robust PEBS	41
Simulated Annealing	85
Differential Evolution	66
Monte Carlo	283

6.2.1.2 Case 2

Uncertainties of the case are depicted in Table 7 and a solid three-phase short-circuit occurs at bus 1 and the fault is cleared by the opening of the line 1-5.

Table 7: Uncertainties Case 2

Line		Parameter Uncertainty (%)		
From Bus	To Bus	R	X	B/2
1	2	10	10	1
1	5	10	10	0
2	4	0.1	0	0
2	5	0	10	0
3	4	1	0	0
4	7	1	0	0
4	9	0	8	5
5	6	0	0	5
6	11	4	1.5	0
6	12	0	1	0
6	13	1	0	2
7	8	0	5	0
7	9	1	0	30
9	10	1	0	0
9	14	0	5	0
10	11	0	5	0
12	13	0.1	0	0
13	14	5	0	5

For the second case, we obtained the following results:

Table 8: Case 2 Results

Method	Critical Clearing Time (ms)
Robust PEBS	281
Simulated Annealing	288
Differential Evolution	285
Monte Carlo	285

Table 9: Case 2 Execution Time

Method	Time (min)
Robust PEBS	48
Simulated Annealing	212
Differential Evolution	193
Monte Carlo	848

Looking at Table 8, we can again verify that our proposed methodology found a close result, without resorting to the demanding Monte Carlo methodology. Likewise, the optimization methods also provided good results at a higher computational cost.

6.2.1.3 Case 3

Uncertainties of case 3 are depicted in Table 10 and a solid three-phase short-circuit occurs at bus 1 and the fault is cleared by the opening of the line 1-5.

Table 10: Uncertainties Case 3

Bus	Parameter Uncertainty (%)	
	P_L	Q_L
2	50	40
3	25	0

For the third case, we obtained the following results:

Table 11: Case 3 Results

Method	Critical Clearing Time (ms)
Robust PEBS	288
Simulated Annealing	297
Differential Evolution	297
Monte Carlo	297

Table 12: Case 3 Execution Time

Method	Time (min)
Robust PEBS	51
Simulated Annealing	48
Differential Evolution	204
Monte Carlo	695

Looking at Table 11, we can again verify that our proposed methodology found a close result, without resorting to the demanding Monte Carlo methodology. Likewise, the optimization methods also provided good results. However, unlike the other cases, the Simulated Annealing routine outperformed the Differential Evolution by a large margin. An explanation for this is that since the loads produce a deeper effect in the critical clearing time, the solution variance is greater. Since the stopping criterion for the latter depends on the variance, a larger number of iterations were required. The first one, on the other hand, is noticeably less affected as we predefined an initial temperature value and at each cooling, fewer solutions are accepted.

It is interesting to observe that in all cases the execution time for the robust PEBS is similar, as our method assumes that all numbers are intervals. Hence, the number of interval operations/interval function evaluations remains unchanged from a small set of uncertainties to a larger set. This particularity of the method can be both advantageous and disadvantageous. For a small set of uncertainties, the robust PEBS can be slower than the optimization algorithms, making the use of the latter preferable. For a larger set, however, the robust PEBS will be faster, making it preferable.

6.3 Sensitivity Analysis

In this section, we test the proposed sensitivity analysis in the IEEE 14 bus system. The test was conducted according to the following sequence: First, the most influential parameters for the transient analysis were selected using our proposed methodology. Afterward, the Monte Carlo method was employed to assess the transient stability of the power system in two situations, one with every parameter having a 2.5% uncertainty, and another with only the most influential parameters having a 2.5% uncertainty.

Applying the proposed sensitivity analysis using the Principal Component Analysis on a solid three-phase short-circuit occurring at bus 1, we obtain the following eigenvalue $\pi_{\max}$ and eigenvector $u_{\max}$:

$$\pi_{\max} = 70679.6954, \quad (6.1)$$

$$u_{\max} = \begin{bmatrix} 1.6632\text{e-}01, & 4.8867\text{e-}01, & 2.9643\text{e-}01, & 1.6692\text{e-}01 \\ 7.0349\text{e-}02, & 2.1272\text{e-}01, & 6.0199\text{e-}02, & 4.8195\text{e-}02 \\ 2.7024\text{e-}02, & 7.6672\text{e-}02, & 5.3426\text{e-}02, & -1.6451\text{e-}04 \\ -1.3603\text{e-}03, & 3.2206\text{e-}04, & -1.0177\text{e-}03, & -4.7789\text{e-}04 \\ -2.1894\text{e-}03, & -3.0006\text{e-}04, & -1.9554\text{e-}04, & -7.7843\text{e-}05 \\ -3.5867\text{e-}04, & -2.0231\text{e-}04, & -6.3914\text{e-}01, & 2.4473\text{e-}02 \\ -1.0077\text{e-}02, & -4.4425\text{e-}02, & -6.0725\text{e-}02, & 8.3989\text{e-}03 \\ 4.6284\text{e-}03, & 2.2480\text{e-}03, & 1.0042\text{e-}03, & 6.4017\text{e-}03 \\ 9.9117\text{e-}03, & 8.0372\text{e-}03, & 1.0102\text{e-}04, & 2.5656\text{e-}04 \\ 1.0176\text{e-}04, & 3.2731\text{e-}01, & 8.3060\text{e-}02, & -1.7829\text{e-}04 \\ 1.3092\text{e-}02, & 7.1127\text{e-}03, & -7.1135\text{e-}03, & -1.4095\text{e-}03 \\ -6.3035\text{e-}04, & -7.5947\text{e-}04, & 2.8452\text{e-}02, & -6.4826\text{e-}03 \\ -2.6461\text{e-}03, & -1.4551\text{e-}02, & -1.1380\text{e-}02, & -3.8519\text{e-}03 \\ -7.1711\text{e-}04, & -7.5909\text{e-}03, & -7.1956\text{e-}04, & -1.2380\text{e-}03 \\ -9.4389\text{e-}04, & 1.4603\text{e-}01, & 9.1610\text{e-}03, & -1.9722\text{e-}05 \\ 1.2623\text{e-}03, & 7.0766\text{e-}04, & -2.6618\text{e-}04 \end{bmatrix}. \quad (6.2)$$

Choosing a 10% threshold, we obtain the following most influential parameters: $P_{L_2}, P_{L_3}, P_{L_4}, P_{L_5}, P_{L_9}, X_{1-5}, B/2_{1-5}$.

On the other hand, analyzing the sensitivity matrix combined with the K-Means technique, we found the following most influential parameters: $P_{L_2}, P_{L_3}, P_{L_4}, P_{L_5}, P_{L_9}, X_{1-5}$.

In Table 13, we demonstrate the efficiency of the sensitivity analysis by applying the Monte Carlo method for each set of uncertain parameters: Uncertainty in all parameters (Naïve); uncertainty only in the parameters identified by the sensitivity analysis with the Principal Component Analysis; and uncertainty only in the parameters identified by the sensitivity analysis with the K-Means technique.

Table 13: Sensitivity Analysis Methodology Comparison

Method	Critical Clearing Time (ms)	Execution Time (min)	Error (%)
Naïve (uncertainty in all parameters)	306.8	207	—
Principal Component Analysis	309.3	21	0.81
Sensitivity Matrix Analysis	309.5	19	0.88

From the results, it is clear that the sensitivity analysis using both the Principal Component Analysis and the K-Means technique presents a great performance gain with an insignificant error introduction. After all, the assessment was executed 10 times faster and introduced error was less than 1%.

7 CONCLUSION

The aim of this dissertation has been to develop methodologies to study power systems with uncertainties. In particular, fast methodologies to assess in real-time the transient stability. As a result, two methodologies were proposed along the text. The first one is based on the theory of energy functions and employs a robust version of the PEBS method to determine the critical clearing time. We modeled the uncertainties as intervals and applied the theory of interval arithmetic to obtain an interval version of the method. An advantage from employing intervals instead of statistics was the possibility of developing closed formulas that can be directly evaluated without resorting to expensive simulations.

The second methodology was two optimization algorithms to assess the transient stability of the power system with uncertainties more rapidly than using the Monte Carlo method and with similar precision. Specifically, we advocated the use of the Simulated Annealing and Differential Evolution optimizations algorithms. These methods were preferred since they provide a global search for the optimization problem and are derivative-free techniques. A better estimate for the critical clearing time without the hassle of numerically approximating the objective function derivative was therefore possible.

Given the inherent complexity and size of modern power systems, a method to search the most influential parameters for the transient stability assessment was also developed. The method combines a sensitivity analysis with both a clustering technique called K-Means and the Principal Component Analysis to screen the influential/non-influential parameters. For testing the proposed methodologies, a single-machine and a multi-machine system were selected to verify each method's effectiveness. The results demonstrated that all methodologies provided a reliable result and were faster than the Monte Carlo method as well. A compromise between accuracy and speed was also shown, as the robust PEBS surpassed both optimization methods but achieved a less accurate result.

The results also exhibited the sensitivity analysis efficacy when detecting the most influential parameters using the Principal Component Analysis and the K-Means technique. The error between the critical clearing time estimate using the uncertainties in all parameters and in only the most influential parameters—selected by the sensitivity analysis—was less than 1%. Simulation time was 10 times faster, nevertheless.

This work provides a range of options for future research as each developed method can be further improved. From the computational side, the use of more optimized interval arithmetic libraries/modules can improve the speed and accuracy of the robust PEBS method. Except for the Simulated Annealing method—which is serial by nature—the proposed methods can take advantage of parallel computation. In the Differential Evolution method, for example, each computer thread could be associated with a population member,

while in the robust PEBS method the summations present in its formulation could be parallelized for faster evaluation of the interval energy expressions. From the theoretical side, further studies in the theory of interval differential equation dynamics can be carried out. An implementation of a non-reduced network robust PEBS is also a possible extension of this work.

BIBLIOGRAPHY

- [1] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, T. Van Cutsem, C. Canizares, N. Hatziaargyriou, D. Hill, V. Vittal, A. Stankovic, and C. Taylor, “Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions”, *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387–1401, 2004, ISSN: 0885-8950. DOI: [10.1109/TPWRS.2004.825981](https://doi.org/10.1109/TPWRS.2004.825981).
- [2] T. Athay, R. Podmore, and S. Virmani, “A practical method for the direct analysis of transient stability”, *IEEE Transactions on Power Apparatus and Systems*, no. 2, pp. 573–584, 1979.
- [3] N. G. Bretas and L. F. C. Alberto, “Transient stability analysis of power systems: robustness with respect to parameter uncertainties”, in *Power Engineering Society Winter Meeting, 2002. IEEE*, IEEE, vol. 2, 2002, pp. 1105–1112.
- [4] Y. Bard, *Nonlinear Parameter Estimation*. Academic Press, 1974, ISBN: 0120782502.
- [5] J. B. MacQueen, “Some Methods for Classification and Analysis of MultiVariate Observations”, in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1, University of California Press, 1967, pp. 281–297.
- [6] F. Immler and C. Traut, “The flow of ODEs”, in *International Conference on Interactive Theorem Proving*, Springer, 2016, pp. 184–199.
- [7] R. L. V. Fortulan, “Análise de segurança dinâmica em sistemas elétricos de potência via classificadores BCU”, Trabalho de Conclusão de Curso de Graduação em Engenharia Elétrica com ênfase em Eletrônica, Universidade de São Paulo, 2016.
- [8] D. A. Suprunenko, *Jacobi matrix*, in *Encyclopedia of Mathematics*. [Online]. Available: http://www.encyclopediaofmath.org/index.php?title=Jacobi_matrix&oldid=29168 (visited on 01/29/2019).
- [9] A. Monticelli, *State estimation in electric power systems: a generalized approach*, 1st ed. Springer Science & Business Media, 1999.
- [10] J. J. Grainger and W. D. Stevenson, *Power system analysis*. McGraw-Hill New York, 1994, vol. 621.
- [11] R. A. Serway and J. W. Jewett, *Physics for scientists and engineers with modern physics*. Cengage learning, 2018.
- [12] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*, ser. EPRI power system engineering series. McGraw-Hill, 1994, ISBN: 9780070359581.
- [13] G. Kron, *Tensor Analysis of Networks*, ser. General Electric series. Macdonald, 1965.

- [14] S. Markov, “Calculus for interval functions of a real variable”, *Computing*, vol. 22, no. 4, pp. 325–337, 1979.
- [15] R. E. Moore, “Interval analysis. series in automatic computation”, *Englewood Cliff: NJ Prentice-Hall*, 1966.
- [16] J. Cruz, “Constraint reasoning for differential models”, in *Proceedings of the 2005 conference on Constraint Reasoning for Differential Models*, IOS Press, 2005, pp. 1–216.
- [17] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to interval analysis*. Siam, 2009, vol. 110.
- [18] E. Hansen and G. W. Walster, *Global optimization using interval analysis: revised and expanded*. CRC Press, 2003, vol. 264.
- [19] A. Neumaier, *Interval Methods for Systems of Equations*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1990. DOI: [10.1017/CBO9780511526473](https://doi.org/10.1017/CBO9780511526473).
- [20] R. Krawczyk, “Interval operators and fixed intervals”, *Lecture Notes in Computer Science*, vol. 212 LNCS, pp. 81–94, 1986. DOI: [10.1007/3-540-16437-5_8](https://doi.org/10.1007/3-540-16437-5_8).
- [21] R. L. Burden and J. D. Faires, *Numerical Analysis*. Cengage Learning, 2010, ISBN: 9780538733519.
- [22] E. de Weerd, “Interval Analysis: Contributions to static and dynamic optimization”, PhD thesis, Technische Universiteit Delf, Oct. 1, 2010.
- [23] R. Grimshaw, *Nonlinear Ordinary Differential Equations*. CRC Press, Mar. 6, 1991, 336 pp., ISBN: 0849386071.
- [24] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss, “Validated solutions of initial value problems for ordinary differential equations”, *Applied Mathematics and Computation*, vol. 105, no. 1, pp. 21–68, 1999.
- [25] O. Bouissou, A. Chapoutot, and A. Djoudi, “Enclosing temporal evolution of dynamical systems using numerical methods”, in *NASA Formal Methods Symposium*, Springer, 2013, pp. 108–123.
- [26] M. Vescovi, A. Farquhar, and Y. Iwasaki, “Numerical interval simulation: combined qualitative and quantitative simulation to bound behaviors of non-monotonic systems”, in *IJCAI*, 1995, pp. 1806–1813.
- [27] *Interval and Related Software*. [Online]. Available: <http://www.cs.utep.edu/interval-comp/intsoft.html> (visited on 09/27/2018).

-
- [28] H.-D. Chiang, F. F. Wu, and P. P. Varaiya, “Foundations of the potential energy boundary surface method for power system transient stability analysis”, *IEEE Transactions on Circuits and Systems*, vol. 35, no. 6, pp. 712–728, 1988, ISSN: 00984094. DOI: [10.1109/31.1808](https://doi.org/10.1109/31.1808).
  - [29] M. A. Pai, *Energy Function Analysis for Power System Stability*. Springer US, 1989. DOI: [10.1007/978-1-4613-1635-0](https://doi.org/10.1007/978-1-4613-1635-0).
  - [30] N. Kakimoto, Y. Ohsawa, and M. Hayashi, “Transient stability analysis of electric power system via Lure-Type Lyapunov function, Parts I and II”, *Trans. IEE of Japan*, vol. 98, no. 516, 1978.
  - [31] A. H. El-Abiad and K. Nagappan, “Transient stability regions of multimachine power systems”, *IEEE Transactions on Power Apparatus and Systems*, no. 2, pp. 169–179, 1966.
  - [32] N. G. Bretas and L. F. C. Alberto, *Estabilidade transitória em sistemas eletroenergéticos*. Edusp, 2000, ISBN: 8585205318.
  - [33] P. W. Sauer, M. A. Pai, and J. H. Chow, *Power system dynamics and stability : with synchrophasor measurement and power system toolbox*, English. Hoboken, NJ, USA: John Wiley & Sons Ltd, 2018, vol. Second edition. ISBN: 9781119355779.
  - [34] H.-d. Chiang, F. F. F. F. Wu, and P. P. P. Varaiya, “A BCU method for direct analysis of power system transient stability”, *IEEE Transactions on Power Systems*, vol. 9, no. 3, pp. 1194–1208, 1994, ISSN: 08858950. DOI: [10.1109/59.336079](https://doi.org/10.1109/59.336079).
  - [35] E. Chiodo, F. Gagliardi, and D. Lauria, “Probabilistic approach to transient stability evaluation”, *IEE Proceedings - Generation transmission and distribution*, vol. 141, no. 5, pp. 537–544, 1994.
  - [36] K. Timko, A. Bose, and P. Anderson, “Monte carlo simulation of power system stability”, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-102, no. 10, pp. 3453–3459, 1983, ISSN: 0018-9510. DOI: [10.1109/TPAS.1983.317843](https://doi.org/10.1109/TPAS.1983.317843).
  - [37] S. Ayasun, Y. Liang, and C. O. Nwankpa, “A sensitivity approach for computation of the probability density function of critical clearing time and probability of stability in power system transient stability analysis”, *Applied Mathematics and Computation*, vol. 176, no. 2, pp. 563–576, 2006, ISSN: 00963003. DOI: [10.1016/j.amc.2005.10.003](https://doi.org/10.1016/j.amc.2005.10.003).
  - [38] E. Chiodo, “Transient stability evaluation of multimachine power systems: a probabilistic approach based upon the extended equal area criterion”, *IEE Proceedings - Generation, Transmission and Distribution*, vol. 141, no. 6, p. 545, 1994, ISSN: 13502360. DOI: [10.1049/ip-gtd:19941380](https://doi.org/10.1049/ip-gtd:19941380).

-
- [39] H. M. Rodrigues, L. F. C. Alberto, and N. G. Bretas, “Uniform Invariance Principle and Synchronization. Robustness with Respect to Parameter Variation”, *Journal of Differential Equations*, vol. 169, no. 1, pp. 228–254, Jan. 2001. DOI: [10.1006/jdeq.2000.3902](https://doi.org/10.1006/jdeq.2000.3902).
  - [40] R. Billinton and P. R. S. Kuruganty, “A probabilistic index for transient stability”, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-99, no. 1, pp. 195–206, 1980, ISSN: 00189510. DOI: [10.1109/TPAS.1980.319629](https://doi.org/10.1109/TPAS.1980.319629).
  - [41] —, “Probabilistic assessment of transient stability in a practical multimachine system”, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 7, pp. 3634–3641, 1981, ISSN: 0018-9510. DOI: [10.1109/TPAS.1981.316657](https://doi.org/10.1109/TPAS.1981.316657).
  - [42] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing”, *Science*, vol. 220, no. 4598, pp. 671–680, May 1983. DOI: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671).
  - [43] S. Kirkpatrick, “Optimization by simulated annealing: Quantitative studies”, *Journal of Statistical Physics*, vol. 34, no. 5-6, pp. 975–986, Mar. 1984. DOI: [10.1007/bf01009452](https://doi.org/10.1007/bf01009452).
  - [44] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines”, *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, Jun. 1953. DOI: [10.1063/1.1699114](https://doi.org/10.1063/1.1699114).
  - [45] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1992, ISBN: 0-521-43108-5.
  - [46] L. D. Landau and E. M. Lifshitz, *Statistical Physics*, Third Edition, L. D. Landau and E. M. Lifshitz, Eds. Oxford: Butterworth-Heinemann, 1980, ISBN: 978-0-08-057046-4. DOI: <https://doi.org/10.1016/B978-0-08-057046-4.50005-1>.
  - [47] A. Corana, M. Marchesi, C. Martini, and S. Ridella, “Minimizing multimodal functions of continuous variables with the “simulated annealing” algorithm”, *ACM Transactions on Mathematical Software*, vol. 13, no. 3, pp. 262–280, Sep. 1987. DOI: [10.1145/29380.29864](https://doi.org/10.1145/29380.29864).
  - [48] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
  - [49] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution, A Practical Approach to Global Optimization*, ser. Natural Computing Series. Springer-Verlag, 2005. DOI: [10.1007/3-540-31306-0](https://doi.org/10.1007/3-540-31306-0).

-
- [50] R. Storn, *Differential Evolution (DE) for Continuous Function Optimization (an algorithm by Kenneth Price and Rainer Storn)*. [Online]. Available: <http://www1.icsi.berkeley.edu/~storn/code.html> (visited on 11/26/2018).
  - [51] E. Jones, T. Oliphant, P. Peterson, *et al.*, *SciPy: Open source scientific tools for Python*, 2001–. [Online]. Available: <http://www.scipy.org/> (visited on 01/29/2019).
  - [52] M. Jamil and X. S. Yang, “A literature survey of benchmark functions for global optimisation problems”, *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, p. 150, 2013. DOI: [10.1504/ijmmno.2013.055204](https://doi.org/10.1504/ijmmno.2013.055204).
  - [53] S. Vajda, P. Valko, and T. Turanyi, “Principal component analysis of kinetic models”, *International Journal of Chemical Kinetics*, vol. 17, no. 1, pp. 55–81, 1985.
  - [54] R. Christie, *14 Bus Power Flow Test Case*. [Online]. Available: https://www2.ee.washington.edu/research/pstca/pf14/pg_tca14bus.htm (visited on 08/17/2018).
  - [55] S. Taschini, *PyInterval – Interval Arithmetic in Python*, 2017. [Online]. Available: <https://github.com/taschini/pyinterval> (visited on 01/01/2018).
  - [56] W. Navidi, *Statistics for Engineers and Scientists*, Third edition. McGraw-Hill Science/Engineering/Math, Jan. 27, 2010.
  - [57] W. Oberle, “Monte Carlo Simulations: Number of Iterations and Accuracy”, Army Research Lab Aberdeen Proving Ground MD Weapons and Materials Research Directorate, Tech. Rep., 2015.

Appendix

APPENDIX A – APPENDIX

A.1 Estimating the Number of Monte Carlo Iterations

Suppose a Monte Carlo simulation was performed with n iterations, so that we have obtained a sample $x_1, x_2, \dots, x_n$ with size n from a X distribution/population with mean μ and variance σ^2 . The expected value (or mean) of the sample mean $\bar{x}$ is then [56]:

$$E(\bar{x}) = E\left(\frac{x_1 + x_2 + \dots + x_n}{n}\right). \quad (\text{A.1})$$

Since the expected value operator is linear, we can write:

$$E(\bar{x}) = \frac{1}{n} \cdot [E(x_1) + E(x_2) + \dots + E(x_n)]. \quad (\text{A.2})$$

Replacing $E(x_i) = \mu$, $i = 1, 2, \dots, n$, in above formula yields:

$$E(\bar{x}) = \frac{1}{n} \cdot \underbrace{[\mu + \dots + \mu]}_{n \text{ times}} = \frac{1}{n} \cdot n\mu = \mu. \quad (\text{A.3})$$

Therefore, the expected value of the sample mean is the same as of X [56].

We can proceed in a similar manner to find the variance of the sample mean:

$$\text{Var}(\bar{x}) = \text{Var}\left(\frac{x_1 + x_2 + \dots + x_n}{n}\right). \quad (\text{A.4})$$

Rewriting (A.4), we obtain:

$$\text{Var}(\bar{x}) = \text{Var}\left(\frac{1}{n}x_1 + \frac{1}{n}x_2 + \dots + \frac{1}{n}x_n\right). \quad (\text{A.5})$$

Applying Bienaymé formula in (A.5) yields:

$$\text{Var}(\bar{x}) = \frac{1}{n^2}\text{Var}(x_1) + \frac{1}{n^2}\text{Var}(x_2) + \dots + \frac{1}{n^2}\text{Var}(x_n). \quad (\text{A.6})$$

Since $\text{Var}(x_i) = \sigma^2$, $i = 1, 2, \dots, n$, we obtain the equation as follows:

$$\text{Var}(\bar{x}) = \frac{1}{n^2} \cdot \underbrace{[\sigma^2 + \dots + \sigma^2]}_{n \text{ times}} = \frac{1}{n^2} \cdot n\sigma^2 = \frac{\sigma^2}{n}. \quad (\text{A.7})$$

This result indicates that the variance of the sample mean is inversely proportional to the sample size. To find a value for n , we proceed as follows.

The central limit theorem establishes that for a larger n , $\bar{x}$ follows a normal distribution—with mean and variance as deduced before—and [57]:

$$P\left(-z_{\alpha/2} < \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} < z_{\alpha/2}\right) \approx 1 - \alpha, \quad (\text{A.8})$$

where $100 \cdot (1 - \alpha)\%$ is called the confidence level, and $z_{\alpha/2}$ is known as z -critical value. Solving (A.8) for μ results in the following confidence interval:

$$\bar{x} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}} < \mu < \bar{x} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}}. \quad (\text{A.9})$$

Since we do not know the real variance of X , we instead use the sample variance^I s and modify (A.9) accordingly:

$$\bar{x} - z_{\alpha/2} \frac{s}{\sqrt{n}} < \mu < \bar{x} + z_{\alpha/2} \frac{s}{\sqrt{n}}. \quad (\text{A.10})$$

Subtracting $\bar{x}$ in (A.10) yields:

$$-z_{\alpha/2} \frac{s}{\sqrt{n}} < \mu - \bar{x} < z_{\alpha/2} \frac{s}{\sqrt{n}}. \quad (\text{A.11})$$

The middle term of (A.11) is the error for the mean, denoted by ϵ . Thus, the maximum error is:

$$\epsilon = z_{\alpha/2} \frac{s}{\sqrt{n}}. \quad (\text{A.12})$$

Solving (A.12) for n , we obtain as follows:

$$n = \sqrt{\frac{s \cdot z_{\alpha/2}}{\bar{x} \cdot \epsilon}}. \quad (\text{A.13})$$

Equation (A.13) allows us, for a given error tolerance and confidence level, evaluate the necessary number of iterations in a Monte Carlo simulation. For a small n , however, equation (A.13) must be modified as $\frac{\bar{x} - \mu}{s/\sqrt{n}}$ will follow a Student's t -distribution, and is given as follows :

$$n = \sqrt{\frac{s \cdot t_{\alpha/2, n-1}}{\bar{x} \cdot \epsilon}}, \quad (\text{A.14})$$

where $t_{\alpha/2, n-1}$ is the t -critical value for a Student's t -distribution with $n - 1$ degrees of freedom.

From a practical perspective, it is not viable to estimate first $\bar{x}$, and s to only then calculate n . Instead, we employ (A.13) and (A.14) iteratively as follows:

1. At each m iterations, evaluate $\bar{x}$ and s for the sample;
2. If $m < T$, evaluate^{II} n from (A.14);
3. Else, evaluate n from (A.13);
4. If the number of total iterations exceeds n stop, else continue the process and repeat step 1.

---

^I The unbiased estimator for the sample variance is $s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$.

^{II} T represents a minimum size from which a sample is considered large.